NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

R31825

ALGORITHMS FOR COMPUTER AIDED
DESIGN OF DIGITAL FILTERS

by

Thomas H. Rich

December 1988

Thesis Advisor: Robert Strum

Approved for public release; distribution is unlimited

T242293

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b OFFICE SYMBOL (If applicable) 62	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code) Monterey, California 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification) ALGORITHMS FOR COMPUTER AIDED DESIGN OF DIGITAL FILTERS

12. PERSONAL AUTHOR(S)
RICH, Thomas H.

13a. TYPE OF REPORT	13b TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1988 December	57

16 SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	polynomial substitution; analog-digital design; digital-digital design

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The algorithms necessary for the computer aided design of digital filters from lowpass prototypes has been developed and compared, then implemented in a computer program to aid in the instruction of digital filter design.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL STRUM, Robert D.	22b TELEPHONE (Include Area Code) 408-646-3451 22c OFFICE SYMBOL 62St

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

UNCLASSIFIED

Approved for public release; distribution is unlimited

ALGORITHMS FOR COMPUTER AIDED DESIGN OF DIGITAL FILTERS

by

Thomas H. Rich
Captain, United States Marine Corps
BSME, Rensselaer Polytechnic Institute, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1988

ABSTRACT

The algorithms necessary for the computer aided design of digital filters from lowpass prototypes have been developed and compared, then implemented in a computer program to aid in the instruction of digital filter design.

Thesis
R37825
C.1

TABLE OF CONTENTS

I. DEVELOPMENT OF GENERAL ALGORITHMS..... 1

A. INTRODUCTION..... 1

B. GENERAL DISCUSSION..... 4

II. DEVELOPMENT OF SPECIFIC TRANSFORMATIONS.....13

A. INTRODUCTION.....13

B. BILINEAL TRANSFORMATION.....13

C. LOWPASS TO LOWPASS DIGITAL TRANSFORMATION.....16

D. LOWPASS TO HIGHPASS DIGITAL TRANSFORMATION.....19

E. LOWPASS TO BANDSTOP DIGITAL TRANSFORMATION.....20

F. LOWPASS TO BANDPASS DIGITAL TRANSFORMATION.....23

G. LOWPASS TO LOWPASS ANALOG TRANSFORMATION.....24

H. LOWPASS TO HIGHPASS ANALOG TRANSFORMATION.....25

I. LOWPASS TO BANDPASS ANALOG TRANSFORMATION.....26

J. LOWPASS TO BANDSTOP ANALOG TRANSFORMATION.....28

K. GENERAL OBSERVATIONS.....29

L. IMPLEMENTATION.....32

LIST OF REFERENCES.....34

APPENDIX: DFCADD FORTRAN PROGRAM.....35

INITIAL DISTRIBUTION LIST.....52

I. DEVELOPMENT OF GENERAL ALGORITHMS

A. INTRODUCTION

The procedure for the design of lowpass, highpass, bandpass, and bandstop digital filters from lowpass prototypes is well established, and is based on transforming the prototype frequency response to meet desired characteristics. This transformation can be accomplished in either the analog (s) domain or the digital (z) domain (Figure 1).

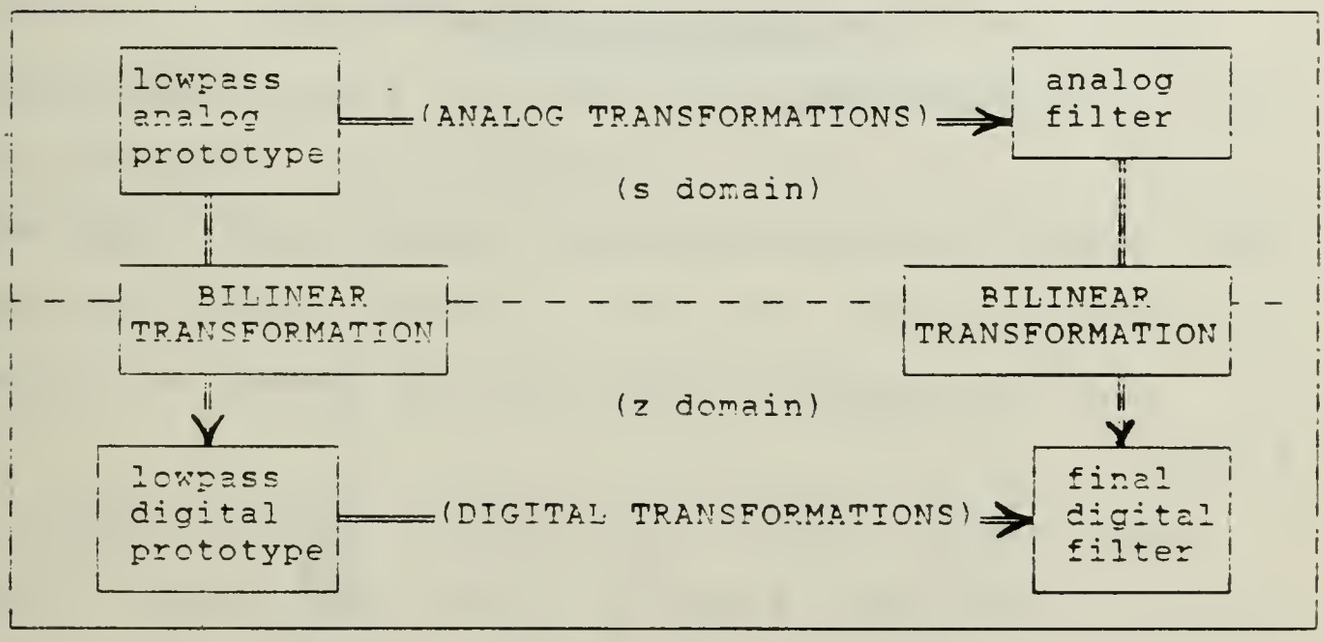

Figure 1. Two methods for digital filter design. Note that each of the horizontal arrows actually represent four different transformations:
1) lowpass prototype to lowpass filter
2) lowpass prototype to highpass filter
3) lowpass prototype to bandpass filter
4) lowpass prototype to bandstop filter

Therefore, the two classical design techniques, analog-digital and digital-digital design, are based on doing the frequency response transformations in their respective analog or digital domains. Since the theory for the standard filter types (Butterworth, Chebyshev, and elliptic) is developed in the analog or 's' domain, the two methods for designing digital filters differ basically in the sequence of the design steps. In the analog-digital method the analog prototype is transformed and then digitized, while in the digital-digital method the analog prototype is digitized and then transformed.

Each of the transformations (arrows in Figure 1) involves substituting a function of 's' or 'z' for the appropriate variable in the filter transfer function. The purpose of this thesis is to develop efficient algorithms to carry out these transformations and then to implement these algorithms in a computer program to aid in the design of digital filters.

Computer-aided design of digital filters offers a significant advantage since it allows the digital filter coefficients to be calculated quickly and accurately. The accuracy of the filter coefficients is an especially significant point since the frequency response of the final filter is very sensitive to small inaccuracies in the calculated coefficients. As an example, compare the frequency response for the filters in Figure 2 and 3. Figure

2 shows the frequency response magnitude for a 1/2-dB ripple Chebyshev bandpass filter, and Figure 3 shows the frequency response when one of the coefficients has been changed by 1/1000th.

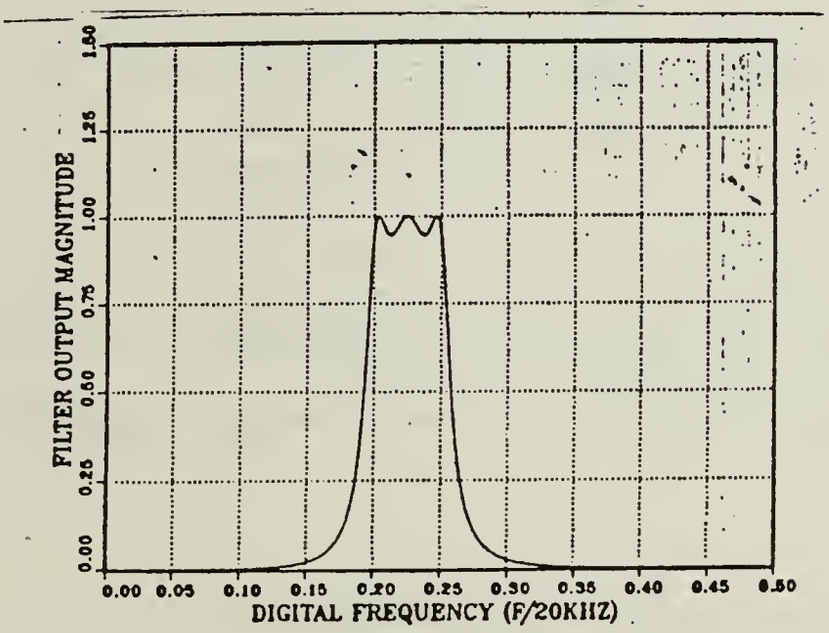

Figure 2. Chebyshev Bandpass Filter

$$H(z) = \frac{.00229(z^4 - 1)^3}{z^6 - .87775z^5 + 2.80076z^4 - 1.5389z^3 + 2.46277z^2 - .67494z + .67529}$$

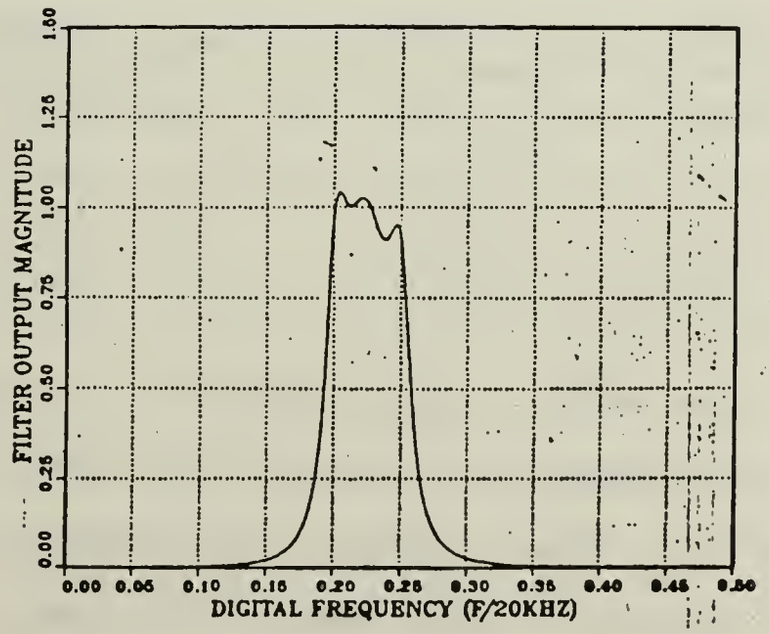

Figure 3. Same Filter With Denominator Coefficient of z changed to -.67594

B. GENERAL DISCUSSION

Each of the transformations involves substituting a ratio of two polynomials of s or z into the filter transfer function which is also a ratio of two polynomials in s or z, namely

$$H(y)=\left.\frac{Y(y)}{U(y)}\right|_{y=\frac{p(x)}{q(x)}} \qquad (1.1)$$

The variables 'x' and 'y' can be either 's' or 'z' depending on the transformation to be performed. Therefore, we have either

$$H(y)=\left.\frac{a_0 + a_1 y + a_2 y^2 + \ldots + a_m y^m}{b_0 + b_1 y + b_2 y^2 + \ldots + b_n y^n}\right|_{y=\frac{p(x)}{q(x)}} \qquad (1.2)$$

or

$$H(x)=\frac{a_0 + a_1 \frac{p(x)}{q(x)} + a_2 \frac{(p(x))^2}{(q(x))^2} + \ldots + a_m \frac{(p(x))^m}{(q(x))^m}}{b_0 + b_1 \frac{p(x)}{q(x)} + b_1 \frac{(p(x))^2}{(q(x))^2} + \ldots + b_n \frac{(p(x))^n}{(q(x))^n}} \quad . \qquad (1.3)$$

In general, the order of the numerator 'm', is less than the order of the denominator 'n' (except for the transformations in the digital domain where the result of the bilinear transformation is to make n=m). We can generalize the

results, however, by assuming that the order or the numerator polynomial Y(x) is equal to the order of the denominator polynomial U(x) and making the additional coefficients $a_{m+1}, a_{m+2}, \ldots, a_n$ equal to zero. Then multiplying both the numerator and denominator of the above equation by $(g(x))^n$ gives

$$H(x) = \frac{a_0 (g(x))^n + a_1 p(x)(g(x))^{n-1} + \ldots + a_n (p(x))^n}{b_0 (g(x))^n + b_1 p(x)(g(x))^{n-1} + \ldots + b_n (p(x))^n} \qquad (1.4)$$

or, in a different form,

$$H(x) = \frac{\sum_{i=0}^{i=n} a_i \, p(x)^i g(x)^{n-i}}{\sum_{i=0}^{i=n} b_i \, p(x)^i g(x)^{n-i}} \qquad (1.5)$$

Now with the numerator and denominator of the same form, we can use the same transformation for both the numerator and denominator polynomials of the filter transfer function, namely,

$$a'(x) = a_0' + a_1' x + a_2' x^2 + \ldots + a_n' x^n = \sum_{i=0}^{i=n} a_i \, p(x)^i g(x)^{n-i} \qquad (1.6)$$

In Equation 1.6, the a_i are the coefficients of the old numerator or denominator polynomials and a_i' are the new, transformed numerator or denominator polynomial coefficients. Since the new coefficients a_i', are a linear combination of

the old coefficients a_i, and the coefficients of p(x) and q(x), this suggests the use of a matrix transformation;

$$\mathbf{a}' = \mathbf{a}\mathbf{A}(n) \tag{1.7}$$

where

a is the coefficient row vector for the old numerator or denominator polynomial of length=n+1.

$$\mathbf{a} = [a_0 \; a_1 \; a_2 \; . \; . \; . \; a_n]$$

a' is the coefficient row vector for the new numerator or denominator polynomial of length=n'+1.

$$\mathbf{a}' = [a'_0 \; a'_1 \; a'_2 \; . \; . \; . \; a_{n'}]$$

A(n) is the transformation matrix with dimensions (n+1)x(n'+1). The elements of **A**(n) depend upon the coefficients of p(x) and q(x). Note that if the rows and columns of **A**(n) are numbered 0 to n', then

$$a_j' = \sum_{i=0}^{i=n} a_i A(n)_{ij} \tag{1.8}$$

(where $A(n)_{ij}$ denotes the element in row i, column j of matrix **A**(n).)

n is the order of the transfer function before the transformation

n' is the order of the transfer function after the transformation. Note that n' equals n times the highest order of p(x) and q(x).

For example, when n=1 and n'=2, p(x) and q(x) are second order and Equation 1.7 becomes,

$$\begin{bmatrix} a_0 & a_1 \end{bmatrix} \begin{bmatrix} A(2)_{00} & A(2)_{01} & A(2)_{02} \\ A(2)_{10} & A(2)_{11} & A(2)_{12} \end{bmatrix} = \begin{bmatrix} a0' & a1' & a2' \end{bmatrix}. \tag{1.9}$$

In the case of the lowpass digital, highpass digital, and bilinear transformations, the two functions p(x) and g(x) are first order. Professor P.H. Moose of the Naval Postgraduate School of Monterey has developed the matrix A(n) for these two cases by expanding Equation 1.6 using the binomial theorem (Reference 1). This method cannot be used for transformations involving higher order polynomials such as the bandpass and bandstop transformations and it is difficult to develop because intricate manipulation of the binomial coefficients are involved. The binomial expansion approach, therefore, was not pursued.

A general method has been found that develops the required transformation matrices in an iterative manner. In other words, the matrix used to transform a numerator or denominator filter polynomial of order n is developed from the matrix for the polynomial of order n-1. This approach is easy to implement in a computer program and can be used for any of the polynomial substitutions needed. The next chapter is concerned with developing the transformation matrices for each specific transformation. However, as an introduction to the method used, consider the generic case where both p(x) and g(x) are second-order polynomials. Since all of the analog and digital transformations involve substitutions with polynomials of second order or less, this second-order generic case will be all that is needed to develop the specific transformations later.

With $p(x)=ax^2 + bx + c$, $g(x) = dx^2 + ex + f$, and substituting into Equation 1.6 for the first order case of n=1 (remembering that with p(x) and g(x) of second order, the resulting polynomial will be second order), we get

$$a'(x) = a_0' + a_1'x + a_2'x^2 = a_0(dx^2 + ex + f) + a_1(ax^2+bx+c) \tag{1.10}$$

Using Equation 1.8 or 1.9 it is easy to pick out the matrix coefficients $A(1)_{ij}$ (i=0,1; j=0,1,2) for the first order generic case. The coefficients are,

$$\mathbf{A}(1) = \begin{bmatrix} f & e & d \\ c & b & a \end{bmatrix} \tag{1.11}$$

Now with n=2 (n'=4) substituting into Equation 1.6 gives,

$$a'(x) = a_0' + a_1'x + a_2'x^2 + a_3'x^3 + a_4'x^4 \tag{1.12}$$
$$= a_0(dx^2+ex+f)^2 + a_1(ax^2+bx+c)(dx^2+ex+f) + a_2(ax^2+bx+c)^2$$

and the matrix **A**(2) is;

$$\mathbf{A}(2) = \begin{bmatrix} f^2 & 2ef & 2df+e^2 & 2de & d^2 \\ fc & ec+bf & cd+fa+eb & db+ea & da \\ c^2 & 2bc & 2ac+b^2 & 2ab & a \end{bmatrix} \tag{1.13}$$

To see how this matrix is derived from the previous one, consider the first row of the matrices in Equations 1.11 and 1.13. For both matrices, this row is multiplied by the a_0 coefficient in the old polynomial. For n=1 we have $a_0 p(x)$

and we pick out the coefficients of the powers of x (x^j) to get the matrix elements $A(n)_{0,j}$. For n=2, we have $a_0(p(x))^2$ and we do the same thing only now we have the additional factor of p(x). The matrix elements can be calculated by hand in this fashion but it requires a lot of algebra. A simpler method is to use the matrix elements of the previous matrix and get the appropriate elements by convolving coefficients. For example, consider the sequence of elements in row 0 (the top row) of A(1), namely

$$f \quad e \quad d \tag{1.14}$$

To get the elements for row 0 of A(2), we need to multiply (convolve coefficients) by another p(x). Therefore, with rows numbered 0 to n, and columns number 0 to n', we have the following.

For row 0 column 0, the appropriate matrix element is

$$A(2)_{00} = \begin{matrix} d & e & f \\ f & e & d \end{matrix} = f^2 \tag{1.15}$$

For row 0 column 1, the appropriate matrix element is

$$A(2)_{01} = \begin{matrix} & d & e & f \\ f & e & d & \end{matrix} = 2fe \tag{1.16}$$

For row 0 column 2, the appropriate matrix element is

$$A(2)_{02} = \begin{matrix} d & d & e & f \\ f & e & d & \end{matrix} = 2fd+e^2 \tag{1.17}$$

For row 0 column 3, the appropriate matrix element is

$$A(2)_{03} = \begin{matrix} & d & e & f \\ f & e & d & \end{matrix} = 2de \qquad (1.18)$$

For row 0 column 4, the appropriate matrix element is

$$A(2)_{04} = \begin{matrix} & d & e & f \\ f & e & d & \end{matrix} = d^2 \qquad (1.19)$$

This convolution of the coefficients of the old matrix with the coefficients of p(x) can be carried out for every row of the old matrix to obtain all the rows of the new matrix except the last row. In the case of the last row, we are multiplying by an additional g(x) rather than an additional p(x) and so must obtain the last row of the new matrix by convolving the coefficients of the last row of the old matrix with the coefficients of g(x).

In summary we have,

(i) for row 0 to row n-1:

-row i of A(n) = row i of A(n-1) * coefficients of p(x) (1.20)

(ii) for row n:

-row n of A(n) = row n-1 of A(n-1) * coefficients of g(x) (1.21)

where * indicates linear convolution of the coefficients.

To see how this method of convolving coefficients works, consider multiplying two polynomials:

$$f1(x) = f2(x)f3(x) \qquad (1.22)$$

Since 'x' is the independent variable, no matter what letter we use, the polynomials will remain the same. Substituting 'z^{-1}' for 'x' in the above equation gives us,

$$f1(z^{-1}) = f2(z^{-1})f3(z^{-1}) \tag{1.23}$$

and we now have the familiar form where we can either multiply the two functions of 'z' or convolve the sequences found by taking the inverse z-transform. For $f2(z^{-1})$ and $f3(z^{-1})$ we find

$$\begin{aligned} f2(z^{-1}) &= a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n} \\ f3(z^{-1}) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_m z^{-m} \end{aligned} \tag{1.24}$$

It is now clear that we can find the coefficients of the product f1(x) by convolving the sequence $\{a_n\}$ with the sequence $\{b_m\}$ to produce the sequence $\{c_q\}$ where,

$$\begin{aligned} \{a_n\} &= \{a_0, a_1, a_2, a_3, \ldots, a_n\} \\ \{b_m\} &= \{b_0, b_1, b_2, b_3, \ldots, b_m\} \end{aligned}$$

and with $q = m + n$ (1.25)

$$\{c_q\} = \{c_0, c_1, c_2, c_3, \ldots, c_q\}$$

Then by taking the z-transform of the sequence $\{c_q\}$ and substituting 'x' for 'z^{-1}' we get,

$$f1(x) = c_0 + c_1 x + \ldots + c_q x^q \tag{1.26}$$

We now have a simple straight-forward method of developing the necessary transformation matrices for digital filter design by picking out the first-order transformation matrix using Equations 1.6 and 1.11, and then using the convolution of coefficients method iteratively on the rows of each successive transformation matrix with Equations 1.20 and 1.21 until the appropriate matrix **A**(n) has been generated. The next chapter will develop the transformation matrices for each of the following transformations:

- bilinear transformation
- digital lowpass prototype to digital lowpass filter
- digital lowpass prototype to digital highpass filter
- digital lowpass prototype to digital bandpass filter
- digital lowpass prototype to digital bandstop filter
- analog lowpass prototype to analog lowpass filter
- analog lowpass prototype to analog highpass filter
- analog lowpass prototype to analog bandpass filter
- analog lowpass prototype to analog bandstop filter.

We will see that some of these transformations are similar and consequently, we will not have to develop separate transformation matrices for each case.

II. DEVELOPMENT OF SPECIFIC TRANSFORMATIONS

A. INTRODUCTION

The individual transformations are discussed in the following sections. The specific form of the transformations used in each case are from Reference 2. Since the final result will be to develop a computer program to assist in the calculation of digital filter coefficients, confining the specific transformations to the form in Reference 2, will allow the computer program to be used in conjunction with Reference 2 as an aid in designing digital filters.

B. BILINEAR TRANSFORMATION

The bilinear transformation may be used to transform an analog transfer function, H(s), to a digital transfer function H(z). The transformation from H(s) to H(z) is described by

$$H(z) = H(z)\Big|_{s=\frac{z-1}{z+1}} \qquad (2.1)$$

Referring to Equation 1.1, p(x)=z-1 and g(x)=z+1. Since both of the transformation polynomials p(x) and g(x) are first order, the bilinear transformation will not change the order of the transfer function. Therefore, for an analog transfer function of order 'n', the transformation matrix **A(n)** will be

(n+1) by (n+1). From Equation 1.6, the numerator or denominator polynomial of the transfer function H(z) is given by

$$a'(z) = a_0' + a_1'z + \ldots + a_n'z^n = \sum_{i=0}^{i=n} a_i (z-1)^i (z+1)^{n-i} \qquad (2.2)$$

where the unprimed coefficients, a_i, come from the analog transfer function's numerator or denominator polynomial. In matrix form, with n=1 and referring to Equation 1.11, the first order matrix A(1) for the bilinear transformation is

$$A(1) = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \qquad (2.3)$$

Using the convolution of coefficients method developed in Chapter 1, the transformation matrix A(n+1) can be developed from the matrix A(n). The results for the bilinear transformation matrices A(1) through A(5) are given in Table 1.

TABLE 1
FIRST-ORDER TO FIFTH-ORDER BILINEAR TRANSFORMATION MATRICES

$$\mathbf{A}(1) = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$\mathbf{A}(2) = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 0 & 1 \\ 1 & -2 & 1 \end{bmatrix}$$

$$\mathbf{A}(3) = \begin{bmatrix} 1 & 3 & 3 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & 3 & -3 & 1 \end{bmatrix}$$

$$\mathbf{A}(4) = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ -1 & -2 & 0 & 2 & 1 \\ 1 & 0 & -2 & 0 & 1 \\ -1 & 2 & 0 & -2 & 1 \\ 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

$$\mathbf{A}(5) = \begin{bmatrix} 1 & 5 & 10 & 10 & 5 & 1 \\ -1 & -3 & -2 & 2 & 3 & 1 \\ 1 & 1 & -2 & -2 & 1 & 1 \\ -1 & 1 & 2 & -2 & -1 & 1 \\ 1 & -3 & 2 & 2 & -3 & 1 \\ -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix}$$

Referring to Table 1 and keeping in mind the convolution of coefficients method that produced the matrices, the following relations are apparent and will be useful in computer implementation.

$$A(n)_{0,j} \text{ (the first row)} = \binom{n}{j} \quad \text{for } j=0,1,\ldots,n \tag{2.4}$$

where $\binom{n}{j}$ indicates the binomial coefficient

$$A(n)_{i,0} \text{ (the first column)} = (-1)^i \quad \text{for } i=0,1,\ldots,n \tag{2.5}$$

$$A(n)_{i,n} \text{ (the last column)} = 1 \quad \text{for } i=0,1,\ldots,n \tag{2.6}$$

$$A(n)_{(n-i),j} = (-1)^{n-j} A(n)_{i,j} \quad \text{for } i=0,1\ldots,\text{integer}(n/2)+1 \tag{2.7}$$

$$A(n)_{i,j} = A(n-1)_{i,j} + A(n-1)_{i,(j-1)} \quad \text{for } i=0,1,\ldots,n-1 \quad j=1,2,\ldots,n-1 \tag{2.8}$$

Notice that the values of the matrix elements for the bilinear transformation are whole numbers. This is important from a numerical accuracy standpoint since whole numbers can be expressed precisely in floating point form. This means that the bilinear transformation will cause a loss of accuracy of the original coefficients only because of the n+1 multiplications and additions used to calculate each new digital coefficient.

C. LOWPASS TO LOWPASS DIGITAL TRANSFORMATION

The lowpass to lowpass digital transformation is used to transform a lowpass prototype digital filter **H(z)** into a lowpass digital filter H(z). The polynomial substitution

used is

$$H(z) = H_p(z)\Big|_{z=\frac{z-\alpha}{1-\alpha z}} \qquad (2.9)$$

where the subscript p indicates the prototype digital filter transfer function. The design constant α is determined [References 2 and 3] by

$$\alpha = \frac{\sin(\theta_c/2 - \theta_c'/2)}{\sin(\theta_c/2 - \theta_c''/2)} \qquad (2.10)$$

with θ_c = the prototype critical frequency (usually pi/2) and θ_c' = the desired critical frequency of the lowpass filter. Again, referring to Equation 1.1 with $p(x) = z-\alpha$ and $g(z) = 1-\alpha z$, the lowpass digital transformation matrix will not change the order of the transfer function and the transformation matrix **A**(n) will be (n+1) by (n+1). Using Equations 1.6 and 1.11 as before, the first order transformation matrix **A**(1) for the lowpass digital transformation is:

$$A(1) = \begin{bmatrix} 1 & -\alpha \\ -\alpha & 1 \end{bmatrix} \qquad (2.11)$$

Once again using the convolution of coefficients method, the higher-order transformation matrixes **A**(n) are developed from the lower-order matrix **A**(n-1). The results are shown in Table 2 for the first-order to fifth-order transformation matrices.

TABLE 2
FIRST-ORDER TO FIFTH-ORDER LOWPASS DIGITAL
TRANSFORMATION MATRICES

$$\mathbf{A}(1) = \begin{bmatrix} 1 & -\alpha \\ -\alpha & 1 \end{bmatrix}$$

$$\mathbf{A}(2) = \begin{bmatrix} 1 & -2\alpha & \alpha^2 \\ -\alpha & (1+\alpha^2) & -\alpha \\ \alpha^2 & -2\alpha & 1 \end{bmatrix}$$

$$\mathbf{A}(3) = \begin{bmatrix} 1 & -3\alpha & 3\alpha^2 & -\alpha^3 \\ -\alpha & (1+2\alpha^2) & (-2\alpha-\alpha^3) & \alpha^2 \\ \alpha^2 & (-2\alpha-\alpha^3) & (1+2\alpha^2) & -\alpha \\ -\alpha^3 & 3\alpha^2 & -3\alpha & 1 \end{bmatrix}$$

$$\mathbf{A}(4) = \begin{bmatrix} 1 & -4\alpha & 6\alpha^2 & -4\alpha^3 & \alpha^4 \\ -\alpha & (1+3\alpha^2) & (-3\alpha-3\alpha^3) & (3\alpha^2+\alpha^4) & -\alpha^3 \\ \alpha^2 & (-2\alpha-2\alpha^3) & (1+4\alpha^2+\alpha^4) & (-2\alpha-2\alpha^3) & \alpha^2 \\ -\alpha^3 & (3\alpha^2+\alpha^4) & (-3\alpha-3\alpha^3) & (1+3\alpha^2) & -\alpha \\ \alpha^4 & -4\alpha 3 & 6\alpha^2 & -4\alpha & 1 \end{bmatrix}$$

$$\mathbf{A}(5) = \begin{bmatrix} 1 & -5\alpha & 10\alpha^2 & -10\alpha^3 & 5\alpha^4 & -\alpha^5 \\ -\alpha & (1+3\alpha^2) & (-4\alpha-6\alpha^3) & (6\alpha^2+\alpha^4) & (-4\alpha^3-\alpha^5) & \alpha 4 \\ \alpha^2 & (-2\alpha-3\alpha^3) & (1+6\alpha^2+3\alpha^4) & (-3\alpha-6\alpha^3-\alpha^5) & (3\alpha^2+2\alpha^4) & -\alpha^3 \\ -\alpha^3 & (3\alpha^2+2\alpha^4) & (-3\alpha-6\alpha^2-\alpha^5) & (1+6\alpha^2+3\alpha^4) & (-2\alpha-3\alpha^3) & \alpha 2 \\ \alpha^4 & (-4\alpha^3-\alpha^5) & (6\alpha^2+4\alpha^4) & (-4\alpha-6\alpha^3) & (1+4\alpha^2) & -\alpha \\ -\alpha^5 & 5\alpha^4 & -10\alpha^3 & 10\alpha^2 & -5\alpha & 1 \end{bmatrix}$$

Referring to Table 2, the following relations are observed and will be used in the computer implementation.

$$A(n)_{0j} \text{ (the first row)} = \binom{n}{j} (-\alpha)^j \quad \text{for } j=0,1,\ldots,n \tag{2.12}$$

$$A(n)_{i0} \text{ (the first column)} = (-\alpha)^i \quad \text{for } i=0,1,\ldots,n \tag{2.13}$$

$$A(n)_{ij} = A(n)_{(n-i)(n-j)} \quad \text{for } i,j=0,1,\ldots,n \tag{2.14}$$

$$A(n)_{ij} = A(n-1)_{(i-1)(j-1)} + (-\alpha)A(n-1)_{(i-1)j} \tag{2.15}$$
$$\text{for } i=1,2,\ldots,n-1 \quad j=1,2,\ldots,n$$

Note that the elements of the lowpass digital transformation matrices are not whole numbers as in the bilinear transformation case, but are polynomials of the design constant α. Numerical error is introduced in this case by both the n+1 additions or multiplications, and the error in calculating the matrix elements that multiply the original coefficients.

D. LOWPASS TO HIGHPASS DIGITAL TRANSFORMATION

The lowpass to highpass digital transformation is used to transform a lowpass prototype to a highpass filter. The polynomial substitution for this transformation is:

$$H(z) = H_p(z)\Big|_{z = -\frac{z-\alpha}{1-\alpha z}} \tag{2.16}$$

Since this transformation is the same as that for the lowpass to lowpass digital transformation except for the minus sign,

we can make the substitution of '-z' for 'z' in the original prototype transfer function and then use the same transformation matrices as the lowpass to lowpass case. For the lowpass to highpass transformation, the design constant α is (References 2 and 3)

$$\alpha = \frac{\cos(\Theta_c/2 - \Theta_c'/2)}{\cos(\Theta_c/2 + \Theta_c'/2)} \tag{2.17}$$

with Θc - the prototype critical frequency (usually pi/2) and $\Theta c'$ - the desired critical frequency of the highpass filter.

E. LOWPASS TO BANDSTOP DIGITAL TRANSFORMATION

The polynomial substitution which transforms a digital lowpass prototype to a bandstop digital filter is (References 2 and 3):

$$H(z) = H_p(z)\Big|_{z = \frac{z^2 - \frac{2\alpha}{1+k} z + \frac{1-k}{1+k}}{1 - \frac{2\alpha}{1+k} z + \frac{1-k}{1+k} z^2}} \tag{2.18}$$

with

$$\alpha = \frac{\cos(\Theta_u'/2 + \Theta_l'/2)}{\cos(\Theta_u'/2 - \Theta_l'/2)} \tag{2.19}$$

and

$$K = \tan(\Theta_c/2)\tan(\Theta_u'/2 - \Theta_l'/2) \tag{2.20}$$

with once again, Θ_c = the prototype critical frequency (usually pi/2), Θ_u' = the desired upper critical digital

frequency, and Θ_l' = the desired low critical digital frequency.

By letting $b = 2\alpha/(k+1)$ and $c=(1-k)/(1+k)$, Equation 2.18 can be simplified to

$$H(z) = H_p(z)\Big|_{z = \frac{z^2 - bz + c}{1 - bz + cz^2}} \tag{2.21}$$

Again, using Equations 1.6 and 1.11 the first-order transformation matrix is

$$\mathbf{A}(1) = \begin{bmatrix} 1 & -b & c \\ c & -b & 1 \end{bmatrix} \tag{2.22}$$

Note that since the substitution polynomials are second order, this transformation will double the order of the original prototype filter. Therefore, the transformation matrix $\mathbf{A}(n)$ will be (n+1) by (2n+1). The higher order transformation matricies are again developed recursively and the results for the first-order to third-order transformations are given in Table 3.

TABLE 3
FIRST-ORDER TO THIRD-ORDER BANDSTOP
DIGITAL TRANSFORMATION MATRICES

$$\mathbf{A}(1) = \begin{bmatrix} 1 & -b & c \\ c & -b & 1 \end{bmatrix}$$

$$\mathbf{A}(2) = \begin{bmatrix} 1 & -2b & (b^2+2c) & -2bc & c^2 \\ c & (-b-bc) & (1+b^2+c^2) & (-b-bc) & c \\ c^2 & -2bc & (b^2+2c) & -2c & 1 \end{bmatrix}$$

$$\mathbf{A}(3) = \begin{bmatrix} 1 & -3c & (3b^2+3c) & (-b^3-6bc) & \dots \\ c & (-2bc-b) & (1+2b^2+bc+2c) & (-b^3-2b-2bc-2bc^2) & \dots \\ c^2 & (-2bc-bc^2) & (c^3+2b^2c+2c+b^2) & \dots & \\ c^3 & -3c^2b & (3b^2c+3c^2) & \dots & \end{bmatrix}$$

(Redundant entries not shown for **A**(3) due to space limitations. Entries not shown can be found using Equation 2.25)

Notice the increased amount of calculation needed to determine the matrix elements for the bandstop case. Each matrix element is a function of 'b' and 'c' which are themselves functions of the design constants α and k. Table 3 only gives the transformation matrices for first- to third-order. However, to illustrate the amount of calculation required, the matrix element $A(5)_{25}$ is

$$A(5)_{25} = (-12bc^2-6bc^3-8b^3c-6bc-6b^3-3b-3bc^4-6b^3c^2-b^5) \quad (2.23)$$

Since each coefficient in the final filter is found by multiplying n+1 matrix elements by the n+1 original

coefficients and then adding the n+1 products, the accuracy of the design constants and the original coefficients can be critical in the calculation of the final filter coefficients.

The symmetry relations for the bandstop transformation are:

$$A(n)_{i0} \text{ (the first column)} = c^i \quad \text{for } i=0,1,\ldots,n \tag{2.24}$$

$$A(n)_{ij} = A(n)_{(n-i)(2n-j)} \quad \text{for } i=0,1,\ldots,n \quad j=0,1,\ldots,2n \tag{2.25}$$

$$A(n)_{ij} = A(n-1)_{ij} - bA(n-1)_{i(j-1)} + cA(n-1)_{i(j-2)} \quad \text{for } i=0,1,\ldots,n-1 \quad j=0,1,\ldots,2n \tag{2.26}$$

F. LOWPASS TO BANDPASS DIGITAL TRANSFORMATION

Again, the polynomial substitution which transforms the lowpass prototype to a bandpass filter is

$$H(z) = H_p(z)\Big|_{z = -\dfrac{z^2 - \frac{2\alpha k}{k+1} z + \frac{k-1}{k+1}}{1 - \frac{2\alpha k}{k+1} z + \frac{k-1}{k+1} z^2}} \tag{2.27}$$

where in this case $k = \tan(\Theta/2)\cot(\Theta_u'/2 - \Theta_l'/2)$ and α is found from the same expression as the bandstop case. With the simplification of $b=2\alpha k/(k+1)$ and $c=(k-1)/(k+1)$, Equation 2.27 becomes

$$H(z) = H_p(z)\Big|_{z = -\frac{z^2 - bz + c}{1 - bz + cz^2}} \qquad (2.28)$$

Since this is the same form as the bandstop transformation, we can use the matrices previously developed if '-z' is first substituted for 'z' as was done in the lowpass and highpass situation.

G. LOWPASS TO LOWPASS ANALOG TRANSFORMATION

The lowpass to lowpass analog transformation is used to transform an analog lowpass prototype to a lowpass analog filter. The polynomial substitution is a simple one, namely

$$H(s) = H_p(s)\Big|_{s = s/w_c} \qquad (2.29)$$

where w_c is the prewarped critical frequency ($w_c = \tan(\Theta_c/2)$ and Θ_c is the desired digital critical frequency ($\Theta_c = 2(pi)f_c/f_s$). This transformation matrix is found to be

$$\mathbf{A}(n) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1/w_c & 0 & \cdots & 0 \\ 0 & 0 & 1/w_c^2 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 1/w_c^n \end{bmatrix}$$

or $\mathbf{A}(n) = \text{diag}(1, 1/w_c, 1/w_c^2, \ldots, 1/w_c^n)$ throughout. (2.30)

Note that in this case, each new filter coefficient depends only on multiplying the old coefficient of s^1 by $(w_c)^{-1}$. Therefore, any loss of accuracy due to the transformation is due to this one multiplication and the error introduced in calculation of the matrix element $(w_c)^{-1}$.

H. LOWPASS TO HIGHPASS ANALOG TRANSFORMATION

The polynomial substitution for the lowpass to highpass transformation is

$$H(s) = H_p(s)\Big|_{s = w_c/s} \tag{2.31}$$

Again using Equations 1.6 and 1.11, the first-order transformation matrix is

$$A(1) = \begin{bmatrix} 0 & 1 \\ w_c & 0 \end{bmatrix} \tag{2.32}$$

The method of convolution of coefficients can be used to get the higher-order transformation matrices. However, notice that Equation 2.32 is the same matrix as the lowpass to lowpass transformation with the reciprocal taken of each non zero element, and then each row reversed. This can be shown to be the case in general, so that the lowpass to highpass transformation matrix is given by

$$\mathbf{A}(n) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & w_c & 0 \\ \vdots & & & & \\ 0 & w_c^{n-1} & \cdots & 0 & 0 \\ w_c^{n} & 0 & \cdots & 0 & 0 \end{bmatrix} \qquad (2.33)$$

Again, each new filter coefficient depends only on one multiplication although the calculation of the matrix element itself will require calculations involving raising the design constant, w_c, to the $(n-1)^{th}$ power where i is the power of s in the final filter transfer function.

I. LOWPASS TO BANDPASS ANALOG TRANSFORMATION

The polynomial substitution for transforming a lowpass analog prototype to a bandpass analog filter is

$$H(s) = H_p(s)\Big|_{s = \frac{s^2 + w_o^2}{Bs}} \qquad (2.34)$$

where wo is defined as the geometric mean of the passband; $w_o = (w_u w_l)^{\frac{1}{2}}$, and B is the width of the passband, $B = w_u - w_l$. (w_u and w_l are the upper and lower critical frequencies respectively). Again using Equations 1.6 and 1.11 and picking out the appropriate matrix elements, the first-order transformation matrix is given by

$$\mathbf{A}(1) = \begin{bmatrix} 0 & B & 0 \\ w_o^2 & 0 & 1 \end{bmatrix} \tag{2.35}$$

The convolution of coefficients technique is again used to generate the higher order transformation matrices with the results for the first-order to third-order matrices listed in Table 4. As with the bandpass digital transformation, the analog bandpass transformation will double the order of the original transfer function since the substitution polynomial is of second order.

TABLE 4
FIRST-ORDER TO THIRD-ORDER BANDPASS
ANALOG TRANSFORMATION MATRICES

$$\mathbf{A}(1) = \begin{bmatrix} 0 & B & 0 \\ w_o^2 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}(2) = \begin{bmatrix} 0 & 0 & B^2 & 0 & 0 \\ 0 & Bw_o^2 & 0 & B & 0 \\ w_o^4 & 0 & 2w_o^2 & 0 & 1 \end{bmatrix}$$

$$\mathbf{A}(3) = \begin{bmatrix} 0 & 0 & 0 & B^3 & 0 & 0 & 0 \\ 0 & 0 & B^2 w_o^2 & 0 & B^2 & 0 & 0 \\ 0 & Bw_o^4 & 0 & 2Bw_o^2 & 0 & B & 0 \\ w_o^6 & 0 & 3w_o^4 & 0 & 3w_o^2 & 0 & 1 \end{bmatrix}$$

Note that each matrix is slightly over half empty and that the nonzero elements are simple functions of the design constants w_o and B.

J. LOWPASS TO BANDSTOP ANALOG TRANSFORMATION

For the analog bandstop transformation, the polynomial substitution is the reciprocal of the bandpass case:

$$H(s) = H_p(s)\Big|_{s = \frac{Bs}{s^2 + w_o^2}} \tag{2.36}$$

Using the same method as before, the first-order transformation matrix is found to be:

$$A(1) = \begin{bmatrix} w_o^2 & 0 & 1 \\ 0 & B & 0 \end{bmatrix} \tag{2.37}$$

Note that this is the same matrix as the bandpass case but with the rows in reverse order. Using the convolution of coefficients technique the higher-order matrices are developed and the results for the first-order to third-order transformation matrices are shown in Table 5. Notice that the reversal of the order of the rows compared with the bandpass case is true in general. Therefore, once again the matrices are over half empty and nonzero elements are simple functions of B and w_o.

TABLE 5
FIRST-ORDER TO THIRD-ORDER BANDSTOP
ANALOG TRANSFORMATION MATRICES

$$\mathbf{A}(1) = \begin{bmatrix} w_o^2 & 0 & 1 \\ 0 & B & 0 \end{bmatrix}$$

$$\mathbf{A}(2) = \begin{bmatrix} w_o^4 & 0 & 2w_o^2 & 0 & 1 \\ 0 & Bw_o^4 & 0 & B & 0 \\ 0 & 0 & B^2 & 0 & 0 \end{bmatrix}$$

$$\mathbf{A}(3) = \begin{bmatrix} w_o^6 & 0 & 2w_o^4 & 0 & 3w_o^{22} & 0 & 1 \\ 0 & Bw_o^4 & 0 & 2Bw_o^2 & 0 & B & 0 \\ 0 & 0 & B^2 w_o^2 & 0 & B^2 & 0 & 0 \\ 0 & 0 & 0 & B^3 & 0 & 0 & 0 \end{bmatrix}$$

K. GENERAL OBSERVATIONS

Now that the various transformation matrices for digital filter design have been developed, it is possible to compare the two methods of digital filter design mentioned earlier and depicted in Figure 1. For this comparison, consider an analog lowpass prototype filter transfer function with numerator of order m, and denominator of order n (m less than or equal to n).

First consider the digital to digital direct design method. Transforming the analog prototype to a digital prototype using the bilinear transformation will require m+1

multiplications and additions per digital prototype coefficient for the numerator polynomial, and n+1 multiplications and additions per digital prototype coefficient for the denominator polynomial. Then, each of the digital to digital transformations will require 2(n+1) multiplications and additions since a result of the bilinear transformation is to make n = m. Therefore, going from an analog prototype to a final digital filter will require (3n+m+4) multiplications and additions. If lowpass digital prototypes are developed prior to the design process, this can be reduced to 2(n+1) since then only the digital to digital transformations will be used in designing the final filter.

Now consider the analog design method for the same case of an m^{th} -order numerator and n^{th} -order denominator lowpass prototype transfer function. For the lowpass and highpass case, each new coefficient will require only 1 multiplication and addition. For the bandpass and bandstop transformations a simple expression for the number of calculations per coefficient is difficult to find, however referring to Tables 4 and 5, the number of multiplications and additions per coefficient is always less than half the order of the polynomial to be transformed, on the average. This gives approximately (n+m)/2 for the bandpass to bandstop analog transformations. Finally the analog filter must be converted to a digital filter using the bilinear transformation which

will require (n+m+2) calculations per coefficient for the highpass and lowpass case and 2(n+m+1) for the bandpass and bandstop case. The total number of calculations in using the analog design method then is (n+m+4) for the high/low pass and (2.5(n+m)+2) for the bandpass/stop. These results are summarized in Table 6.

TABLE 6
NUMBER OF MULTIPLICATIONS AND ADDITIONS REQUIRED PER COEFFICIENT TO FIND THE DIGITAL FILTER COEFFICIENTS FROM A LOWPASS ANALOG PROTOTYPE (WITH AN M^{TH} ORDER NUMERATOR AND N^{TH} ORDER DENOMINATOR)

	Analog Design	
	high/low pass	bandpass/stop
analog transformations	2	.5(n+m)
bilinear transformation	n+m+2	2(n+m)+2
Total	n+m+4	2.5(n+m)+2

	Digital Design	
	high/low pass	bandpass/stop
bilinear transformation	n+m+2	n+m+2m
digital transformations	2(n+1)	2(n+1)
Total	3(n+1)+m+1	3(n+1)+m+1

Referring to Table 6, the advantage of one method over the other depends upon what type of filter is being designed and the order of the prototype transfer functions numerator and denominator. Consider designing a fifth-order low/highpass Chebyshev or Butterworth filter. In this case, the order of the numerator polynomial is zero and the analog method will have a slight advantage of 9 multiplications and additions versus 12 for the digital design with precalculated prototypes (and 19 for digital design without precalculated prototypes). However, if a bandpass/stop elliptic filter was being designed with n=5, then m=4 and the direct digital design method (with precalculated prototypes) has an advantage of 12 multiplications and additions compared to approximately 24 for the analog method.

A final consideration in favor of analog design is the complexity of the elements of the transformation matrices. Since the elements of the digital transformation matrices require much more calculation than the elements of the analog transformation matrices or the bilinear transformation (compare Tables 1-5), the problem of roundoff error is much more serious in the case of the digital to digital design method.

L. IMPLEMENTATION

The final step is to implement the algorithms developed previously in a computer program to assist in the teaching of digital filter design. Since the program is intended to be

used for instructional purposes, the numerical considerations discussed earlier are not considered as important as conceptual simplicity. The direct design method was chosen for implementation as the design constants can be found easily without reference to "prewarping." The completed program, called 'DFCADD' (Digital Filter Computer Aided Direct Design), is contained in Appendix A with its own documentation. The program follows the procedures set forth in Chapter 10 of Reference 2. Prototype coefficients are stored within the program for first-order to fifth-order Butterworth and Chebyshev (1/2,1,&2dB ripples) filters. While second-order to sixth-order elliptic lowpass prototype coefficients are calculated within the program for any combination of 1/2,1,2, and 3 dB passband ripple and -20,-30,-40,-50,-60, and -70 dB stopband attenuation, with the help of a program developed by Professor D.E. Kirk of the Naval Postgraduate School and References 4 and 5. As a final option, the user may enter his/her own prototype coefficients for up to a tenth-order analog prototype. After calculating the digital filter coefficients an option exists for the program to create a data file, titled "filter", with the digital filter magnitude and phase as a function of the digital frequency from zero to pi. This data file can then be used with any plotting routine to generate a plot of the frequency response of the filter.

LIST OF REFERENCES

1. Moose, P.H. of the United States Naval Postgraduate School, Monterey, California, Letter to Strum, R.D. and Kirk, D.E., Subject: Algorithm for Digital-Digital Transformations, 5 June 1987.

2. Strum, R.D. and Kirk, D.E., First Principles of Discrete Systems and Digital Signal Processing, pp. 612-701, Addison-Wesley, 1988.

3. Constantinides, A.G., "Spectral Transformations for Digital Filters, "Proceedings of IEEE, p. 117,(August): 1585-1590.

4. Ludeman, L.C., Fundamentals of Digital Signal Processing, pp. 149-160, Harper and Row, 1986.

5. Antoniou, A., Digital Filters: Analysis and Design, pp. 114-125, McGraw-Hill, 1979.

APPENDIX
DFCADD FORTRAN PROGRAM

FILE: DFCADD FORTRAN A

```
C     ************************************************************** NEW00010
C     * THIS PROGRAM ASSISTS IN THE CALCULATION OF DIGITAL FILTER    * NEW00020
C     * COEFFICIENTS ACCORDING TO THE PROCEDURE SET FORTH IN CHAPTER * NEW00030
C     * 10 (SUMMARIZED ON PG 695) OF - FIRST PRINCIPLES OF DISCRETE  * NEW00040
C     * SYSTEMS AND DIGITAL SIGNAL PROCESSING -  BY R.D. STRUM  AND  * NEW00050
C     * D.E. KIRK (REFERENCE(2) )                                    * NEW00060
C     ************************************************************** NEW00070
      INTEGER ORDER,TYPE,PASS,ORDER2,END,AFORDF                       NEW00080
      REAL KPR,LAM,OMEGA(20),V(20),A0(20),B0(20),B1(20)               NEW00090
      COMPLEX Z,NEWZ,ZN,ZD,HOFZN,HOFZD,HOFZ                          NEW00100
      DOUBLE PRECISION TEMP,SUMR                                      NEW00110
      DOUBLE PRECISION R,S0,A01,A02,B01,B02,B11,B12,H0                NEW00120
      DOUBLE PRECISION A03,A04,B03,B04,B13,B14                        NEW00130
      DOUBLE PRECISION SFREQ,AFC,DFC,ALPHA,PI,PID4,DFCD2              NEW00140
      DOUBLE PRECISION ALC,AUC,DLC,DUC,RK,B,C,DLCD2,DUCD2             NEW00150
      DOUBLE PRECISION APROT,BILINR,DPROT,LPHPTR,BPBSTR,DFLTR         NEW00160
C     ************************************************************** NEW00170
C     FORMAT OF ARRAYS:                                               NEW00180
C        -APROT CONTAINS THE ANALOG PROTOTYPE                         NEW00190
C             APROT(NORD,PWR)                                         NEW00200
C        -BILINR IS THE ANALOG-DIGITAL BILINEAR TRANSFORMATION MATRIX NEW00210
C             BILINR(ORDER,ROW,COL)                                   NEW00220
C        -DPROT IS THE DIGITAL PROTOTYPE FROM APROT AND BILINR        NEW00230
C             DPROT(NORD,PWR)                                         NEW00240
C        -LPHPTR IS THE LP&HP DIGITAL-DIGITAL TRANFORMATION MATRIX    NEW00250
C             LPHPTR(ORDER,ROW,COL)                                   NEW00260
C        -DFLTR IS THE FINAL DIGITAL FILTER                           NEW00270
C             DFLTR(NORD,PWR)                                         NEW00280
C        -BPBSTR IS THE BP&BS DIGITAL-DIGITAL TRANSFORMATION MATRIX   NEW00290
C             BPBSTR(ORDER,ROW,COL)                                   NEW00300
C     ************************************************************** NEW00310
      DIMENSION APROT(2,0:10)                                         NEW00320
      DIMENSION BILINR(10,0:10,0:10)                                  NEW00330
      DIMENSION DPROT(2,0:10)                                         NEW00340
      DIMENSION LPHPTR(10,0:10,0:10)                                  NEW00350
      DIMENSION BPBSTR(10,0:10,0:20)                                  NEW00360
      DIMENSION DFLTR(2,0:20)                                         NEW00370
C     ************************************************************** NEW00380
      OPEN(UNIT=8,FILE='FILTER DATA',STATUS='UNKNOWN')                NEW00390
C     ************************************************************** NEW00400
      PI=3.14159265358979D+00                                         NEW00410
      SPI=3.14159                                                     NEW00420
      PID4=PI/4.0D+00                                                 NEW00430
C     ************************************************************** NEW00440
C     CLEAR APROT PRIOR TO LOADING                                    NEW00450
      DO 1 K=1,2                                                      NEW00460
       DO 2 L=0,10                                                    NEW00470
        APROT(K,L)=0.0D+00                                            NEW00480
2      CONTINUE                                                       NEW00490
1     CONTINUE                                                        NEW00500
C     ************************************************************** NEW00510
C     ************************************************************** NEW00520
C     BEGIN INTERACTION                                               NEW00530
      CALL EXCMS ('CLRSCRN')                                          NEW00540
      WRITE(6,*) 'THIS IS A PROGRAM TO CALCULATE DIGITAL FILTERS USING' NEW00550
      WRITE(6,*) ' THE DIRECT DESIGN METHOD.'                         NEW00560
      WRITE(6,*)                                                      NEW00570
      WRITE(6,*) '************************************************'  NEW00580
      WRITE(6,*) '*WARNING-THIS PROGRAM IS NOT USER FRIENDLY!!!  *'   NEW00590
      WRITE(6,*) '* YOU MUST ENTER VALUES OF THE APPROPRIATE TYPE*'   NEW00600
      WRITE(6,*) '* (REAL OR INTEGER), AND PROPER RANGE          *'   NEW00610
      WRITE(6,*) '************************************************'  NEW00620
      WRITE(6,*)                                                      NEW00630
      WRITE(6,*) '  ** ENTER 1 TO CONTINUE ** '                       NEW00640
      READ(5,*)  LOOK                                                 NEW00650
9     CALL EXCMS ('CLRSCRN')                                          NEW00660
10    WRITE(6,*) 'WHAT TYPE OF FILTER DO YOU WANT?'                   NEW00670
      WRITE(6,*) ' (1-4 ALLOW ONLY UP TO 5TH ORDER PROTOTYPES)'       NEW00680
      WRITE(6,*) '      1-BUTTERWORTH '                               NEW00690
      WRITE(6,*) '      2-CHEBYSHEV WITH 1/2 DB RIPPLE '              NEW00700
      WRITE(6,*) '      3-CHEBYSHEV WITH 1 DB RIPPLE '                NEW00710
      WRITE(6,*) '      4-CHEBYSHEV WITH 2 DB RIPPLE'                 NEW00720
```

FILE: DFCADD FORTRAN A

```
      WRITE(6,*) '      5-ELLIPTIC (ALLOWS UP TO A 6TH ORDER PROTOTYPE)'NEW00730
      WRITE(6,*) '        -PASSBAND RIPPLE 0.5,1.0,2.0,OR 3.0 DB'       NEW00740
      WRITE(6,*) '        -STOPBAND ATTENTUATION -20,-30,-40,-50,-60, ' NEW00750
      WRITE(6,*) '             OR -70 DB   '                            NEW00760
      WRITE(6,*) '      6-OTHER - ALLOWS YOU TO USE UP TO A 10TH ORDER' NEW00770
      WRITE(6,*) '         PROTOTYPE, HOWEVER YOU MUST ENTER THE ANALOG'NEW00780
      WRITE(6,*) '         PROTOTYPE COEFFICIENTS '                     NEW00790
      WRITE(6,*)                                                        NEW00800
      WRITE(6,*) '      ** INTEGER 1-6 **'                           NEW00810
      READ(5,*) TYPE                                                    NEW00820
C     ************************************************************      NEW00830
14    CALL EXCMS ('CLRSCRN')                                            NEW00840
15    WRITE(6,*)                                                        NEW00850
      WRITE(6,*) 'WHAT IS THE ORDER OF THE PROTOTYPE YOU WISH TO USE?'  NEW00860
      WRITE(6,*)                                                        NEW00870
      WRITE(6,*) ' - MUST BE AN INTEGER IN THE RANGE 1-5  UNLESS YOU'   NEW00880
      WRITE(6,*) '     CHOSE TO ENTER YOUR OWN PROTOTYPE (THEN 1-10)'   NEW00890
      WRITE(6,*) '     OR YOU CHOSE AN ELLIPTIC FILTER (THEN 1-6)'      NEW00900
      WRITE(6,*) ' - OR ENTER "0" IF YOU WANT HELP SELECTING THE ORDER' NEW00910
      WRITE(6,*) '     BY CHANGING SPECIFICATION FREQUENCIES TO THE'    NEW00920
      WRITE(6,*) '     CORRESPONDING DIGITAL PROTOTYPE FREQUENCIES '    NEW00930
      WRITE(6,*) '       NOTE: YOU WILL NEED PROTOTYPE FREQUENCY   '    NEW00940
      WRITE(6,*) '     RESPONSE PLOTS LIKE THOSE GIVEN IN REF(2) FOR  ' NEW00950
      WRITE(6,*) '     BUTTERWORTH FILTERS (FIG 10.33, PGS 691&692)  '  NEW00960
      WRITE(6,*) '     AND CHEBYSHEV FILTERS (FIG 10.34, PGS 696-698) ' NEW00970
      READ(5,*) ORDER                                                   NEW00980
      IF((TYPE.GT.4).AND.(TYPE.LT.8))THEN                               NEW00990
       IF(ORDER.EQ.1)THEN                                               NEW01000
        WRITE(6,*) 'ELLIPTIC FILTERS GIVEN FOR 2ND-9TH ORDER ONLY'      NEW01010
        WRITE(6,*)                                                      NEW01020
        GO TO 10                                                        NEW01030
       ENDIF                                                            NEW01040
      ENDIF                                                             NEW01050
C                                                                       NEW01060
      CALL EXCMS ('CLRSCRN')                                            NEW01070
      WRITE(6,*) 'DO YOU WANT TO ENTER SPECIFICATION FREQUENCIES'       NEW01080
      WRITE(6,*) 'AS ANALOG OR DIGITAL FREQUENCIES?'                    NEW01090
      WRITE(6,*) '    1-ANALOG  '                                       NEW01100
      WRITE(6,*) '    2-DIGITAL '                                       NEW01110
      WRITE(6,*) '  ** INTEGER 1 OR 2 ** '                              NEW01120
      READ(5,*) AFORDF                                                  NEW01130
      CALL EXCMS ('CLRSCRN')                                            NEW01140
      IF(AFORDF.EQ.1)THEN                                               NEW01150
        WRITE(6,*) 'WHAT IS THE SAMPLING FREQUENCY IN KHZ?'             NEW01160
        WRITE(6,*) '   ** DECIMAL **'                                   NEW01170
        READ(5,*) SSFREQ                                                NEW01180
        SFREQ=DBLE(SSFREQ)                                              NEW01190
      ENDIF                                                             NEW01200
      CALL EXCMS ('CLRSCRN')                                            NEW01210
      WRITE(6,*)                                                        NEW01220
      WRITE(6,*) 'WHAT TYPE OF PASSBAND DUE YOU WANT?'                  NEW01230
      WRITE(6,*) '    1-LOWPASS'                                        NEW01240
      WRITE(6,*) '    2-HIGHPASS'                                       NEW01250
      WRITE(6,*) '    3-BANDPASS'                                       NEW01260
      WRITE(6,*) '    4-BANDSTOP'                                       NEW01270
      WRITE(6,*)                                                        NEW01280
      WRITE(6,*) '        ** INTEGER 1-4 ** '                           NEW01290
      READ(5,*) PASS                                                    NEW01300
C     FORMULAS FOR HIGHPASS AND LOWPASS                                 NEW01310
      IF((PASS.EQ.1).OR.(PASS.EQ.2))THEN                                NEW01320
       CALL EXCMS ('CLRSCRN')                                           NEW01330
       IF(AFORDF.EQ.1)THEN                                              NEW01340
         WRITE(6,*) 'WHAT IS THE CRITICAL FREQUENCY IN KHZ? '           NEW01350
         WRITE(6,*) '    - NEG 3DB PT FOR BUTTERWORTH  '                NEW01360
         WRITE(6,*) '    - RIPPLE EDGE FOR ELLIPTIC AND CHEBYSHEV'      NEW01370
         WRITE(6,*) '       ** DECIMAL ** '                             NEW01380
         READ(5,*) SAFC                                                 NEW01390
         AFC=DBLE(SAFC)                                                 NEW01400
         DFC=(AFC/SFREQ)*PI*(2.0D+00)                                   NEW01410
       ENDIF                                                            NEW01420
       IF(AFORDF.EQ.2)THEN                                              NEW01430
         WRITE(6,*) 'WHAT IS THE CRITICAL DIGITAL FREQUENCY?'           NEW01440
```

FILE: DFCADD FORTRAN A

```
        APROT(2,0)=0.326890D+00                                          NEW04330
        APROT(2,1)=1.022190D+00                                          NEW04340
        APROT(2,2)=0.737822D+00                                          NEW04350
        APROT(2,3)=1.0D+00                                               NEW04360
       ELSEIF(ORDER.EQ.4)THEN                                            NEW04370
        APROT(1,0)=0.163445D+00                                          NEW04380
        APROT(2,0)=0.205765D+00                                          NEW04390
        APROT(2,1)=0.516798D+00                                          NEW04400
        APROT(2,2)=1.256482D+00                                          NEW04410
        APROT(2,3)=0.716215D+00                                          NEW04420
        APROT(2,4)=1.0D+00                                               NEW04430
       ELSEIF(ORDER.EQ.5)THEN                                            NEW04440
        APROT(1,0)=0.081723D+00                                          NEW04450
        APROT(2,0)=0.081723D+00                                          NEW04460
        APROT(2,1)=0.459349D+00                                          NEW04470
        APROT(2,2)=0.693477D+00                                          NEW04480
        APROT(2,3)=1.499543D+00                                          NEW04490
        APROT(2,4)=0.706461D+00                                          NEW04500
        APROT(2,5)=1.0D+00                                               NEW04510
       ENDIF                                                             NEW04520
      ENDIF                                                              NEW04530
C     APROT(TYPE=5) ARE ELLIPTIC FILTERS                                 NEW04540
C     ELLIPTIC COEFFICIENTS ARE CALCULATED USING A PROGRAM DEVELOPED     NEW04550
C     BY PROFESSOR D. E. KIRK, NAVAL POSTGRADUATE SCHOOL, MONTEREY,      NEW04560
C     CALIFORNIA  93943                                                  NEW04570
      A02=0.0D+00                                                        NEW04580
      B02=0.0D+00                                                        NEW04590
      B12=0.0D+00                                                        NEW04600
      S0=0.0D+00                                                         NEW04610
      IF(TYPE.EQ.5)THEN                                                  NEW04620
        CALL EXCMS ('CLRSCRN')                                           NEW04630
        WRITE(6,*) 'WHAT PASSBAND RIPPLE DO YOU WANT? (IN DB)'           NEW04640
        WRITE(6,*) '*** MUST BE EITHER 0.5, 1.0, 2.0, OR 3.0 ***'        NEW04650
        READ(5,*) ASUBP                                                  NEW04660
        WRITE(6,*)                                                       NEW04670
        WRITE(6,*)                                                       NEW04680
        WRITE(6,*) 'WHAT STOPBAND ATTENUATION DO YOU WANT? (IN DB)'      NEW04690
        WRITE(6,*) '*** MUST BE -20,-30,-40,-50,-60, OR -70 ***'         NEW04700
        WRITE(6,*) '*** DO NOT INCLUDE A DECIMAL ***'                    NEW04710
        READ(5,*) NATTN                                                  NEW04720
        NATTN=-1*NATTN                                                   NEW04730
        ASUBA=REAL(NATTN)                                                NEW04740
        IF(ASUBP.EQ.0.5)THEN                                             NEW04750
          IF(NATTN.EQ.20)THEN                                            NEW04760
            IF(ORDER.EQ.2)SR=2.76261                                     NEW04770
            IF(ORDER.EQ.3)SR=1.42189                                     NEW04780
            IF(ORDER.EQ.4)SR=1.13188                                     NEW04790
            IF(ORDER.EQ.5)SR=1.04465                                     NEW04800
            IF(ORDER.EQ.6)SR=1.01553                                     NEW04810
          ENDIF                                                          NEW04820
          IF(NATTN.EQ.30)THEN                                            NEW04830
            IF(ORDER.EQ.2)SR=4.80880                                     NEW04840
            IF(ORDER.EQ.3)SR=1.92322                                     NEW04850
            IF(ORDER.EQ.4)SR=1.32446                                     NEW04860
            IF(ORDER.EQ.5)SR=1.12912                                     NEW04870
            IF(ORDER.EQ.6)SR=1.05394                                     NEW04880
          ENDIF                                                          NEW04890
          IF(NATTN.EQ.40)THEN                                            NEW04900
            IF(ORDER.EQ.2)SR=8.848925                                    NEW04910
            IF(ORDER.EQ.3)SR=2.71147                                     NEW04920
            IF(ORDER.EQ.4)SR=1.62842                                     NEW04930
            IF(ORDER.EQ.5)SR=1.27264                                     NEW04940
            IF(ORDER.EQ.6)SR=1.12697                                     NEW04950
          ENDIF                                                          NEW04960
          IF(NATTN.EQ.50)THEN                                            NEW04970
            IF(ORDER.EQ.2)SR=15.06069                                    NEW04980
            IF(ORDER.EQ.3)SR=3.90430                                     NEW04990
            IF(ORDER.EQ.4)SR=2.06924                                     NEW05000
            IF(ORDER.EQ.5)SR=1.48469                                     NEW05010
            IF(ORDER.EQ.6)SR=1.24101                                     NEW05020
          ENDIF                                                          NEW05030
          IF(NATTN.EQ.60)THEN                                            NEW05040
```

```
        IF(ORDER.EQ.2)SR=26.76231
        IF(ORDER.EQ.3)SR=5.67937
        IF(ORDER.EQ.4)SR=2.88325
        IF(ORDER.EQ.5)SR=1.77664
        IF(ORDER.EQ.6)SR=1.41138
      ENDIF
      IF(NATTN.EQ.70)THEN
        IF(ORDER.EQ.2)SR=47.58853
        IF(ORDER.EQ.3)SR=8.38124
        IF(ORDER.EQ.4)SR=3.52135
        IF(ORDER.EQ.5)SR=2.16377
        IF(ORDER.EQ.6)SR=1.61413
      ENDIF
    ENDIF
    IF(ASUBP.EQ.1.0)THEN
      IF(NATTN.EQ.20)THEN
        IF(ORDER.EQ.2)SR=2.32474
        IF(ORDER.EQ.3)SR=1.38797
        IF(ORDER.EQ.4)SR=1.19109
        IF(ORDER.EQ.5)SR=1.08826
        IF(ORDER.EQ.6)SR=1.05902
      ENDIF
      IF(NATTN.EQ.30)THEN
        IF(ORDER.EQ.2)SR=4.00423
        IF(ORDER.EQ.3)SR=1.73254
        IF(ORDER.EQ.4)SR=1.23141
        IF(ORDER.EQ.5)SR=1.09884
        IF(ORDER.EQ.6)SR=1.13799
      ENDIF
      IF(NATTN.EQ.40)THEN
        IF(ORDER.EQ.2)SR=7.14488
        IF(ORDER.EQ.3)SR=2.42619
        IF(ORDER.EQ.4)SR=1.50549
        IF(ORDER.EQ.5)SR=1.20868
        IF(ORDER.EQ.6)SR=1.19887
      ENDIF
      IF(NATTN.EQ.50)THEN
        IF(ORDER.EQ.2)SR=12.48470
        IF(ORDER.EQ.3)SR=3.46761
        IF(ORDER.EQ.4)SR=2.00129
        IF(ORDER.EQ.5)SR=1.40121
        IF(ORDER.EQ.6)SR=1.29851
      ENDIF
      IF(NATTN.EQ.60)THEN
        IF(ORDER.EQ.2)SR=22.17688
        IF(ORDER.EQ.3)SR=5.82120
        IF(ORDER.EQ.4)SR=2.46179
        IF(ORDER.EQ.5)SR=1.67161
        IF(ORDER.EQ.6)SR=1.34334
      ENDIF
      IF(NATTN.EQ.70)THEN
        IF(ORDER.EQ.2)SR=39.42283
        IF(ORDER.EQ.3)SR=7.30098
        IF(ORDER.EQ.4)SR=3.21982
        IF(ORDER.EQ.5)SR=2.02567
        IF(ORDER.EQ.6)SR=1.55847
      ENDIF
    ENDIF
    IF(ASUBP.EQ.2.0)THEN
      IF(NATTN.EQ.20)THEN
        IF(ORDER.EQ.2)SR=1.84032
        IF(ORDER.EQ.3)SR=1.21801
        IF(ORDER.EQ.4)SR=1.13365
        IF(ORDER.EQ.5)SR=1.01567
        IF(ORDER.EQ.6)SR=1.00447
      ENDIF
      IF(NATTN.EQ.30)THEN
        IF(ORDER.EQ.2)SR=3.28235
        IF(ORDER.EQ.3)SR=1.35690
        IF(ORDER.EQ.4)SR=1.18281
        IF(ORDER.EQ.5)SR=1.14594
        IF(ORDER.EQ.6)SR=1.02460
```

FILE: DFCADD FORTRAN A

```
      ENDIF                                                          NEW05770
      IF(NATTN.EQ.40)THEN                                            NEW05780
        IF(ORDER.EQ.2)SR=5.76107                                     NEW05790
        IF(ORDER.EQ.3)SR=2.13923                                     NEW05800
        IF(ORDER.EQ.4)SR=1.40842                                     NEW05810
        IF(ORDER.EQ.5)SR=1.16811                                     NEW05820
        IF(ORDER.EQ.6)SR=1.07316                                     NEW05830
      ENDIF                                                          NEW05840
      IF(NATTN.EQ.50)THEN                                            NEW05850
        IF(ORDER.EQ.2)SR=10.19181                                    NEW05860
        IF(ORDER.EQ.3)SR=3.04137                                     NEW05870
        IF(ORDER.EQ.4)SR=1.75285                                     NEW05880
        IF(ORDER.EQ.5)SR=1.33243                                     NEW05890
        IF(ORDER.EQ.6)SR=1.15868                                     NEW05900
      ENDIF                                                          NEW05910
      IF(NATTN.EQ.60)THEN                                            NEW05920
        IF(ORDER.EQ.2)SR=18.09398                                    NEW05930
        IF(ORDER.EQ.3)SR=4.39729                                     NEW05940
        IF(ORDER.EQ.4)SR=2.24440                                     NEW05950
        IF(ORDER.EQ.5)SR=1.56860                                     NEW05960
        IF(ORDER.EQ.6)SR=1.28693                                     NEW05970
      ENDIF                                                          NEW05980
      IF(NATTN.EQ.70)THEN                                            NEW05990
        IF(ORDER.EQ.2)SR=32.15951                                    NEW06000
        IF(ORDER.EQ.3)SR=6.40894                                     NEW06010
        IF(ORDER.EQ.4)SR=2.92363                                     NEW06020
        IF(ORDER.EQ.5)SR=1.88906                                     NEW06030
        IF(ORDER.EQ.6)SR=1.46330                                     NEW06040
      ENDIF                                                          NEW06050
    ENDIF                                                            NEW06060
    IF(ASUBP.EQ.3.0)THEN                                             NEW06070
      IF(NATTN.EQ.20)THEN                                            NEW06080
        IF(ORDER.EQ.2)SR=1.73915                                     NEW06090
        IF(ORDER.EQ.3)SR=1.15516                                     NEW06100
        IF(ORDER.EQ.4)SR=1.03853                                     NEW06110
        IF(ORDER.EQ.5)SR=1.00996                                     NEW06120
        IF(ORDER.EQ.6)SR=1.00260                                     NEW06130
      ENDIF                                                          NEW06140
      IF(NATTN.EQ.30)THEN                                            NEW06150
        IF(ORDER.EQ.2)SR=2.90352                                     NEW06160
        IF(ORDER.EQ.3)SR=1.45814                                     NEW06170
        IF(ORDER.EQ.4)SR=1.14542                                     NEW06180
        IF(ORDER.EQ.5)SR=1.05020                                     NEW06190
        IF(ORDER.EQ.6)SR=1.01783                                     NEW06200
      ENDIF                                                          NEW06210
      IF(NATTN.EQ.40)THEN                                            NEW06220
        IF(ORDER.EQ.2)SR=5.05584                                     NEW06230
        IF(ORDER.EQ.3)SR=1.98022                                     NEW06240
        IF(ORDER.EQ.4)SR=1.34663                                     NEW06250
        IF(ORDER.EQ.5)SR=1.13934                                     NEW06260
        IF(ORDER.EQ.6)SR=1.05892                                     NEW06270
      ENDIF                                                          NEW06280
      IF(NATTN.EQ.50)THEN                                            NEW06290
        IF(ORDER.EQ.2)SR=8.93009                                     NEW06300
        IF(ORDER.EQ.3)SR=2.79862                                     NEW06310
        IF(ORDER.EQ.4)SR=1.66148                                     NEW06320
        IF(ORDER.EQ.5)SR=1.28850                                     NEW06330
        IF(ORDER.EQ.6)SR=1.13534                                     NEW06340
      ENDIF                                                          NEW06350
      IF(NATTN.EQ.60)THEN                                            NEW06360
        IF(ORDER.EQ.2)SR=15.84610                                    NEW06370
        IF(ORDER.EQ.3)SR=4.03471                                     NEW06380
        IF(ORDER.EQ.4)SR=2.11596                                     NEW06390
        IF(ORDER.EQ.5)SR=1.50711                                     NEW06400
        IF(ORDER.EQ.6)SR=1.25325                                     NEW06410
      ENDIF                                                          NEW06420
      IF(NATTN.EQ.70)THEN                                            NEW06430
        IF(ORDER.EQ.2)SR=28.15950                                    NEW06440
        IF(ORDER.EQ.3)SR=5.87251                                     NEW06450
        IF(ORDER.EQ.4)SR=2.74749                                     NEW06460
        IF(ORDER.EQ.5)SR=1.80680                                     NEW06470
        IF(ORDER.EQ.6)SR=1.41799                                     NEW06480
```

```
        ENDIF
      ENDIF
      SPI=3.14159
      RK=1.0/SR
      N=ORDER
      EN=N
      KPR=SQRT(1.0-RK**2)
      Q0=0.5*(1.0-SQRT(KPR))/(1.0+SQRT(KPR))
      Q=Q0+2.0*Q0**5+15.0*Q0**9+150.0*Q0**13
      D=(10.0**(0.1*ASUBA)-1.0)/(10.0**(0.1*ASUBP)-1.0)
      LAM=(1.0/(2.0*EN))*ALOG((10.0**(0.05*ASUBP)+1.0)/(10.0**(0.05
     C  *ASUBP)-1.0))
      ISTOP=0
      DENSUM=0.0
      ENSUM=SINH(LAM)
      M=1
17    EM=M
      ENUM=((-1.0)**M)*(Q**(M*(M+1)))*SINH((2.0*EM+1.0)*LAM)
      DEN=2.0*((-1.0)**M)*(Q**(M**2))*COSH(2.0*EM*LAM)
      ENSUM=ENSUM+ENUM
      DENSUM=DENSUM+DEN
      IF(M.GE.2)THEN
        RN=ABS(ENUM/ENSUM)
        RD=ABS(DEN/(1.0+DENSUM))
        IF((RN.LE.1.0E-8).AND.(RD.LE.1.0E-8))THEN
          ISTOP=1
        ENDIF
      ENDIF
      M=M+1
      IF(ISTOP.EQ.0)THEN
        GO TO 17
      ENDIF
      SIGMA0=ABS((2.0*(Q**0.25)*ENSUM)/(1.0+DENSUM))
      W=SQRT((1.0+RK*(SIGMA0**2))*(1.0+(SIGMA0**2/RK)))
      IN=MOD(N,2)
      IF(IN.EQ.0)THEN
        IR=N/2
      ELSE
        IR=(N-1)/2
      ENDIF
      DO 18 I=1,IR
        IF(IN.EQ.0)THEN
          EMU=I-0.5
        ELSE
          EMU=I
        ENDIF
        ISTOP=0
        DENSUM=0.0
        ENSUM=SIN(PI*EMU/EN)
        M=1
19      EM=M
        ENUM=((-1.0)**M)*(Q**(M*(M+1)))*SIN((2.0*EM+1.0)*PI*EMU/EN)
        DEN=2.0*((-1.0)**M)*(Q**(M**2))*COS(2.0*EM*PI*EMU/EN)
        ENSUM=ENSUM+ENUM
        DENSUM=DENSUM+DEN
        IF(M.GE.2)THEN
          RN=ABS(ENUM/ENSUM)
          RD=ABS(DEN/(1.0+DENSUM))
          IF((RN.LE.1.0E-8).AND.(RD.LE.1.0E-8))THEN
            ISTOP=1
          ENDIF
        ENDIF
        M=M+1
        IF(ISTOP.EQ.0)THEN
          GO TO 19
        ENDIF
        OMEGA(I)=(2.0*(Q**0.25)*ENSUM)/(1.0+DENSUM)
        V(I)=SQRT((1.0-RK*OMEGA(I)**2)*(1.0-(OMEGA(I)**2/RK)))
        A0(I)=1.0/(OMEGA(I)**2)
        DN=1.0+(SIGMA0*OMEGA(I))**2
        B0(I)=((SIGMA0*V(I))**2+(OMEGA(I)*W)**2)/(DN**2)
        B1(I)=(2.0*SIGMA0*V(I))/DN
```

FILE: DFCADD FORTRAN A

```
18      CONTINUE                                                        NEW07210
        HO=1.0                                                          NEW07220
        IF(IN.EQ.0)THEN                                                 NEW07230
          DO 20 I=1,IR                                                  NEW07240
            HO=HO*BO(I)/AO(I)                                           NEW07250
20        CONTINUE                                                      NEW07260
          HO=HO*(10.0**(-0.05*ASUBP))                                   NEW07270
        ELSE                                                            NEW07280
          DO 21 I=1,IR                                                  NEW07290
            HO=HO*BO(I)/AO(I)                                           NEW07300
21        CONTINUE                                                      NEW07310
          HO=HO*SIGMAO                                                  NEW07320
        ENDIF                                                           NEW07330
        R=DBLE(SR)                                                      NEW07340
        HO=DBLE(HO)                                                     NEW07350
        A01=DBLE(AO(1))                                                 NEW07360
        B01=DBLE(BO(1))                                                 NEW07370
        B11=DBLE(B1(1))                                                 NEW07380
        IF((ORDER.EQ.2).OR.(ORDER.EQ.3))THEN                            NEW07390
          APROT(1,0)=HO*A01                                             NEW07400
          APROT(1,1)=0.0D+00                                            NEW07410
          APROT(1,2)=HO                                                 NEW07420
          APROT(1,3)=0.0D+00                                            NEW07430
          IF(ORDER.EQ.2)THEN                                            NEW07440
            APROT(2,0)=B01                                              NEW07450
            APROT(2,1)=B11                                              NEW07460
            APROT(2,2)=1.0D+00                                          NEW07470
          ENDIF                                                         NEW07480
          IF(ORDER.EQ.3)THEN                                            NEW07490
            S0=DBLE(SIGMAO)                                             NEW07500
            APROT(2,0)=S0*B01                                           NEW07510
            APROT(2,1)=B01+(B11*S0)                                     NEW07520
            APROT(2,2)=S0+B11                                           NEW07530
            APROT(2,3)=1.0D+00                                          NEW07540
          ENDIF                                                         NEW07550
        ENDIF                                                           NEW07560
        IF((ORDER.EQ.4).OR.(ORDER.EQ.5))THEN                            NEW07570
          A02=DBLE(AO(2))                                               NEW07580
          B02=DBLE(BO(2))                                               NEW07590
          B12=DBLE(B1(2))                                               NEW07600
          APROT(1,0)=HO*A01*A02                                         NEW07610
          APROT(1,1)=0.0D+00                                            NEW07620
          APROT(1,2)=HO*(A01+A02)                                       NEW07630
          APROT(1,3)=0.0D+00                                            NEW07640
          APROT(1,4)=HO                                                 NEW07650
          APROT(1,5)=0.0D+00                                            NEW07660
          IF(ORDER.EQ.4)THEN                                            NEW07670
            APROT(2,0)=B01*B02                                          NEW07680
            APROT(2,1)=(B11*B02)+(B01*B12)                              NEW07690
            APROT(2,2)=B01+B02+(B11*B12)                                NEW07700
            APROT(2,3)=B11+B12                                          NEW07710
            APROT(2,4)=1.0D+00                                          NEW07720
          ENDIF                                                         NEW07730
          IF(ORDER.EQ.5)THEN                                            NEW07740
            S0=DBLE(SIGMAO)                                             NEW07750
            APROT(2,0)=S0*B01*B02                                       NEW07760
            APROT(2,1)=(S0*((B11*B02)+(B01*B12)))+(B01*B02)             NEW07770
            APROT(2,2)=(S0*(B01+B02+(B11*B12)))+(B11*B02)+(B01*B12)     NEW07780
            APROT(2,3)=(S0*(B11+B12))+B01+B02+(B11*B12)                 NEW07790
            APR0T(2,4)=S0+B11+B12                                       NEW07800
            APROT(2,5)=1.0D+00                                          NEW07810
          ENDIF                                                         NEW07820
        ENDIF                                                           NEW07830
        IF((ORDER.EQ.6).OR.(ORDER.EQ.7))THEN                            NEW07840
          A02=DBLE(AO(2))                                               NEW07850
          A03=DBLE(AO(3))                                               NEW07860
          B02=DBLE(BO(2))                                               NEW07870
          B03=DBLE(BO(3))                                               NEW07880
          B12=DBLE(B1(2))                                               NEW07890
          B13=DBLE(B1(3))                                               NEW07900
          APROT(1,0)=HO*A01*A02*A03                                     NEW07910
          APROT(1,1)=0.0D+00                                            NEW07920
```

FILE: DFCADD FORTRAN A

```
          APROT(1,2)=H0*((A02*A03)+(A01*(A02+A03)))                          NEW07930
          APROT(1,3)=0.0D+00                                                 NEW07940
          APROT(1,4)=H0*(A01+A02+A03)                                        NEW07950
          APROT(1,5)=0.0D+00                                                 NEW07960
          APROT(1,6)=H0                                                      NEW07970
          IF((ORDER.EQ.6).OR.(ORDER.EQ.7))THEN                               NEW07980
            APROT(2,0)=B01*B02*B03                                           NEW07990
            APROT(2,1)=(B11*B02*B03)+(B01*((B02*B13)+(B12*B03)))             NEW08000
            APROT(2,2)=(B02*B03)+(B11*((B02*B13)+(B12*B03)))+                NEW08010
     C        (B01*((B02+B03)+(B12*B13)))                                   NEW08020
            APROT(2,3)=B01*(B12+B13)+(B11*((B02+B03)+(B12*B13)))+            NEW08030
     C        ((B02*B13)+(B12*B03))                                         NEW08040
            APROT(2,4)=B01+(B11*(B12+B13))+((B02+B03)+(B12*B13))             NEW08050
            APROT(2,5)=B11+B12+B13                                           NEW08060
            APROT(2,6)=1.0D+00                                               NEW08070
          ENDIF                                                              NEW08080
          IF(ORDER.EQ.7)THEN                                                 NEW08090
            S0=DBLE(SIGMA0)                                                  NEW08100
            APROT(2,7)=1.0D+00                                               NEW08110
            APROT(2,6)=(S0*APROT(2,6))+APROT(2,5)                            NEW08120
            APROT(2,5)=(S0*APROT(2,5))+APROT(2,4)                            NEW08130
            APROT(2,4)=(S0*APROT(2,4))+APROT(2,3)                            NEW08140
            APROT(2,3)=(S0*APROT(2,3))+APROT(2,2)                            NEW08150
            APROT(2,2)=(S0*APROT(2,2))+APROT(2,1)                            NEW08160
            APROT(2,1)=(S0*APROT(2,1))+APROT(2,0)                            NEW08170
            APROT(2,0)=S0*APROT(2,0)                                         NEW08180
          ENDIF                                                              NEW08190
        ENDIF                                                                NEW08200
      ENDIF                                                                  NEW08210
C     NORMALIZE THE ELLIPTIC PROTOTYPES                                      NEW08220
      IF(TYPE.EQ.5)THEN                                                      NEW08230
       DO 25 I=1,2                                                           NEW08240
        DO 26 J=0,ORDER                                                      NEW08250
         APROT(I,J)=APROT(I,J)/((DSQRT(R))**J)                               NEW08260
26      CONTINUE                                                             NEW08270
25     CONTINUE                                                              NEW08280
      ENDIF                                                                  NEW08290
C     APROT(TYPE=6) ALLOWS THE INPUT OF ANY TYPE OF PROTOTYPE FILTER         NEW08300
C       BY INPUTING THE COEFFICIENTS OF S                                    NEW08310
      IF(TYPE.EQ.6)THEN                                                      NEW08320
       WRITE(6,*)                                                            NEW08330
       WRITE(6,*) 'THIS OPTION ALLOWS YOU TO USE ANY ANALOG  '               NEW08340
       WRITE(6,*) ' PROTOTYPE OF ORDER 1-10          '                       NEW08350
       WRITE(6,*) '** COEFFICIENTS MUST BE IN DOUBLE PRECISION '             NEW08360
       WRITE(6,*) '    FORMAT (0.123456789D+00) ** '                         NEW08370
       WRITE(6,*)                                                            NEW08380
       WRITE(6,*) 'INPUT THE NUMERATOR COEFFICIENTS FIRST'                   NEW08390
       DO 50 I=0,ORDER                                                       NEW08400
        WRITE(6,*) 'WHAT IS THE COEFFICIENT OF S**',I,'?'                    NEW08410
        READ(5,*)  APROT(1,I)                                                NEW08420
50     CONTINUE                                                              NEW08430
       WRITE(6,*)                                                            NEW08440
       CALL EXCMS ('CLRSCRN')                                                NEW08450
       WRITE(6,*) 'NOW INPUT THE DENOMINATOR COEFFICIENTS'                   NEW08460
       WRITE(6,*)                                                            NEW08470
       DO 51 J=0,ORDER                                                       NEW08480
        WRITE(6,*) 'WHAT IS THE COEFFICIENT OF S**',J,'?'                    NEW08490
        READ(5,*) APROT(2,J)                                                 NEW08500
51     CONTINUE                                                              NEW08510
      ENDIF                                                                  NEW08520
C     ***********************************************************************NEW08530
C     ***********************************************************************NEW08540
C     DEFINE ELEMENTS OF THE BILINEAR TRANSFORMATION MATRIX                  NEW08550
C     ***********************************************************************NEW08560
C     FOR ALL THE BILINEAR TRANSFORATION MATRICES, THE FIRST COLUMN          NEW08570
C     HAS ALTERNATING +1'S AND -1'S. THE LAST COLUMN HAS ALL +1'S            NEW08580
      DO 60 I=1,ORDER                                                        NEW08590
       DO 61 J=0,I                                                           NEW08600
        BILINR(I,J,0)=(-1.0)**J                                              NEW08610
        BILINR(I,J,I)=1.0                                                    NEW08620
61     CONTINUE                                                              NEW08630
60    CONTINUE                                                               NEW08640
```

FILE: DFCADD FORTRAN A

```
C                                                                       NEW08650
C     FIRST ORDER MATRIX HAS ALREADY BEEN LOADED CONTINUE INTERATION    NEW08660
C     UNTIL YOU HAVE THE APPROPRIATE MATRIX                             NEW08670
      DO 75 K=2,ORDER                                                   NEW08680
       KM1=K-1                                                          NEW08690
C     LOAD ELEMENTS IN INTERIOR COLUMNS, ALL ROWS EXCEPT LAST ONE       NEW08700
       DO 76 L=0,KM1                                                    NEW08710
        DO 77 M=1,KM1                                                   NEW08720
         MM1=M-1                                                        NEW08730
         BILINR(K,L,M)=BILINR(KM1,L,M)+BILINR(KM1,L,MM1)                NEW08740
77      CONTINUE                                                        NEW08750
76     CONTINUE                                                         NEW08760
C     NOW LOAD THE LAST ROW                                             NEW08770
       DO 78 N=1,KM1                                                    NEW08780
        BILINR(K,K,N)=BILINR(K,0,N)*((-1.0D+00)**(K+N))                 NEW08790
78     CONTINUE                                                         NEW08800
75    CONTINUE                                                          NEW08810
C     ******************************************************************NEW08820
C     THE FOLLOWING CLEARS DPROT PRIOR TO LOADING                       NEW08830
      DO 80 K=1,2                                                       NEW08840
       DO 81 L=0,10                                                     NEW08850
        DPROT(K,L)=0.0                                                  NEW08860
81     CONTINUE                                                         NEW08870
80    CONTINUE                                                          NEW08880
C     ******************************************************************NEW08890
C     DIGITIZE THE ANALOG PROTOTYPE USING THE BILINEAR TRANSFORMATION   NEW08900
C     ******************************************************************NEW08910
      DO 100 N=1,2                                                      NEW08920
       DO 101 J=0,ORDER                                                 NEW08930
        SUMR=0.0                                                        NEW08940
        DO 102 I=0,ORDER                                                NEW08950
         TEMP=APROT(N,I)*BILINR(ORDER,I,J)                              NEW08960
         SUMR=SUMR+TEMP                                                 NEW08970
102     CONTINUE                                                        NEW08980
        DPROT(N,J)=SUMR                                                 NEW08990
101    CONTINUE                                                         NEW09000
100   CONTINUE                                                          NEW09010
C     ******************************************************************NEW09020
C     OPTION TO SHOW DIGITAL PROTOTYPE COEFFICIENTS                     NEW09030
      CALL EXCMS ('CLRSCRN')                                            NEW09040
      WRITE(6,*) 'THE DIGITAL PROTOTYPE HAS BEEN COMPUTED. DO YOU'      NEW09050
      WRITE(6,*) 'WISH TO CHECK THE PROTOTYPE COEFFICIENTS?'            NEW09060
      WRITE(6,*) '  ** 1-YES/2-NO **'                                   NEW09070
      READ(5,*) LOOK                                                    NEW09080
      IF(LOOK.EQ.1)THEN                                                 NEW09090
C     MAKE DENOMINATOR COEFFICIENT OF Z**ORDER = 1.0                    NEW09100
       DO 105 K=1,2                                                     NEW09110
        DO 106 L=0,ORDER                                                NEW09120
         DPROT(K,L)=DPROT(K,L)/DPROT(2,ORDER)                           NEW09130
106     CONTINUE                                                        NEW09140
105    CONTINUE                                                         NEW09150
       CALL EXCMS ('CLRSCRN')                                           NEW09160
       WRITE(6,*) 'THE DIGITAL PROTOTYPE NUMERATOR COEFFICIENTS ARE:'   NEW09170
       DO 111 N=ORDER,0,-1                                              NEW09180
        WRITE(6,902) DPROT(1,N),N                                       NEW09190
111     CONTINUE                                                        NEW09200
       WRITE(6,*)                                                       NEW09210
       WRITE(6,*) 'THE DIGITAL PROTOTYPE DENOMINATOR COEFFICIENTS ARE:' NEW09220
       DO 112 M=ORDER,0,-1                                              NEW09230
        WRITE(6,902) DPROT(2,M),M                                       NEW09240
112    CONTINUE                                                         NEW09250
       WRITE(6,*)                                                       NEW09260
       WRITE(6,*) '** 1 TO CONTINUE **'                                 NEW09270
       READ(5,*) LOOK                                                   NEW09280
      ENDIF                                                             NEW09290
C     ******************************************************************NEW09300
C     NOW DO THE DIGITAL TO DIGITAL TRANSFORMATIONS                     NEW09310
C     ******************************************************************NEW09320
      IF((PASS.EQ.1).OR.(PASS.EQ.2))THEN                                NEW09330
C     DEFINE ELEMENTS OF THE LOWPASS AND HIGHPASS DIGITAL-DIGITAL       NEW09340
C     TRANSFORMATION MATRIX                                             NEW09350
C     DEFINE THE ELEMENTS FOR FIRST AND LAST COLUMNS                    NEW09360
```

FILE: DFCADD FORTRAN A

```
       DO 170 I=1,10                                                          NEW09370
        DO 171 J=0,I                                                          NEW09380
         LPHPTR(I,J,0)=(((-1.0)*ALPHA)**J)                                    NEW09390
         IMJ=I-J                                                              NEW09400
         LPHPTR(I,IMJ,I)=LPHPTR(I,J,0)                                        NEW09410
171     CONTINUE                                                              NEW09420
170    CONTINUE                                                               NEW09430
C      FIRST ORDER MATRIX HAS BEEN DEFINED ABOVE                              NEW09440
C      SECOND TO TENTH ORDER BELOW                                            NEW09450
       DO 175 K=2,10                                                          NEW09460
        KM1=K-1                                                               NEW09470
        DO 176 L=0,KM1                                                        NEW09480
         DO 177 M=1,KM1                                                       NEW09490
          MM1=M-1                                                             NEW09500
          LPHPTR(K,L,M)=LPHPTR(KM1,L,M)+((LPHPTR(KM1,L,MM1))*(-1.0D+00)       NEW09510
     C*ALPHA)                                                                 NEW09520
177      CONTINUE                                                             NEW09530
176     CONTINUE                                                              NEW09540
        DO 178 N=1,KM1                                                        NEW09550
         KMN=K-N                                                              NEW09560
         LPHPTR(K,K,N)=LPHPTR(K,0,KMN)                                        NEW09570
178     CONTINUE                                                              NEW09580
175    CONTINUE                                                               NEW09590
C      ***************************************************************        NEW09600
C      CALCULATE THE FINAL FILTER                                             NEW09610
C      FOR HIGHPASS, THE PROTOTYPE MUST SUBSTITUTE (-Z) FOR (Z)               NEW09620
       IF(PASS.EQ.2)THEN                                                      NEW09630
        DO 180 K=1,2                                                          NEW09640
         DO 181 L=0,ORDER                                                     NEW09650
          DPROT(K,L)=DPROT(K,L)*((-1.0)**L)                                   NEW09660
181      CONTINUE                                                             NEW09670
180     CONTINUE                                                              NEW09680
       ENDIF                                                                  NEW09690
       DO 185 K=1,2                                                           NEW09700
        DO 186 L=0,ORDER                                                      NEW09710
         SUMR=0.0D+00                                                         NEW09720
         DO 187 M=0,ORDER                                                     NEW09730
          TEMP=DPROT(K,M)*LPHPTR(ORDER,M,L)                                   NEW09740
          SUMR=SUMR+TEMP                                                      NEW09750
187      CONTINUE                                                             NEW09760
         DFLTR(K,L)=SUMR                                                      NEW09770
186     CONTINUE                                                              NEW09780
185    CONTINUE                                                               NEW09790
      ENDIF                                                                   NEW09800
C      ***************************************************************        NEW09810
C      BANDPASS AND BANDSTOP FORMULAS                                         NEW09820
       IF((PASS.EQ.3).OR.(PASS.EQ.4))THEN                                     NEW09830
C      LOAD THE BANDPASS/BANDSTOP TRANSFORMATION MATRIX                       NEW09840
C      FIRST LOAD THE FIRST AND LAST COLUMNS                                  NEW09850
       DO 225 I=1,10                                                          NEW09860
        DO 226 J=0,I                                                          NEW09870
         BPBSTR(I,J,0)=C**J                                                   NEW09880
         I2=2*I                                                               NEW09890
         IMJ=I-J                                                              NEW09900
         BPBSTR(I,IMJ,I2)=BPBSTR(I,J,0)                                       NEW09910
226     CONTINUE                                                              NEW09920
225    CONTINUE                                                               NEW09930
C      LOAD THE REST OF THE FIRST ORDER MATRIX                                NEW09940
       BPBSTR(1,0,1)=(-1.0D+00)*B                                             NEW09950
       BPBSTR(1,1,1)=(-1.0D+00)*B                                             NEW09960
C      LOAD THE 2ND-10TH ORDER TRANSFORMATION MATRICES                        NEW09970
       DO 230 K=2,10                                                          NEW09980
        K2=2*K                                                                NEW09990
        K2M1=K2-1                                                             NEW10000
        KM1=K-1                                                               NEW10010
        K2M2=K2-2                                                             NEW10020
        KM12=2*KM1                                                            NEW10030
        KM12M1=KM12-1                                                         NEW10040
C      LOAD THE SECOND AND SECOND TO LAST COLUMNS                             NEW10050
        DO 231 L=0,KM1                                                        NEW10060
         BPBSTR(K,L,1)=BPBSTR(KM1,L,1)+(BPBSTR(KM1,L,0)*(-1.0D+00)*B)         NEW10070
         BPBSTR(K,0,K2M1)=(BPBSTR(KM1,0,KM12)*(-1.0D+00)*B)+                  NEW10080
```

FILE: DFCADD FORTRAN A

```
      C     (BPBSTR(KM1,0,KM12M1)*C)                                          NEW10090
            KML=K-L                                                           NEW10100
            BPBSTR(K,KML,K2M1)=BPBSTR(K,L,1)                                  NEW10110
231         CONTINUE                                                          NEW10120
C           NOW LOAD INTERIOR ELEMENTS EXCEPT FOR LAST ROW                    NEW10130
            DO 236 L=0,KM1                                                    NEW10140
             DO 237 M=2,K2M2                                                  NEW10150
              MM1=M-1                                                         NEW10160
              MM2=M-2                                                         NEW10170
             BPBSTR(K,L,M)=BPBSTR(KM1,L,M)+(BPBSTR(KM1,L,MM1)*(-1.0D+00)*     NEW10180
      C       B)+(BPBSTR(KM1,L,MM2)*C)                                        NEW10190
237          CONTINUE                                                         NEW10200
236         CONTINUE                                                          NEW10210
C           NOW FILL THE LAST ROW                                             NEW10220
            DO 238 N=1,K2M1                                                   NEW10230
             K2MN=K2-N                                                        NEW10240
             BPBSTR(K,K,K2MN)=BPBSTR(K,0,N)                                   NEW10250
238         CONTINUE                                                          NEW10260
230        CONTINUE                                                           NEW10270
C          ***************************************************************** NEW10280
C          CALCULATE THE FINAL FILTER                                         NEW10290
C          FOR THE BANDPASS CASE REPLACE (Z) WITH (-Z)                        NEW10300
            IF(PASS.EQ.3)THEN                                                 NEW10310
             DO 240 M=1,2                                                     NEW10320
              DO 241 N=0,ORDER                                                NEW10330
               DPROT(M,N)=DPROT(M,N)*((-1.0D+00)**N)                          NEW10340
241           CONTINUE                                                        NEW10350
240          CONTINUE                                                         NEW10360
            ENDIF                                                             NEW10370
           ORDER2=2*ORDER                                                     NEW10380
           DO 245 K=1,2                                                       NEW10390
            DO 246 L=0,ORDER2                                                 NEW10400
             SUMR=0.0D+00                                                     NEW10410
             DO 247 M=0,ORDER                                                 NEW10420
              TEMP=DPROT(K,M)*BPBSTR(ORDER,M,L)                               NEW10430
              SUMR=SUMR+TEMP                                                  NEW10440
247          CONTINUE                                                         NEW10450
             DFLTR(K,L)=SUMR                                                  NEW10460
246         CONTINUE                                                          NEW10470
245        CONTINUE                                                           NEW10480
           ENDIF                                                              NEW10490
C          ****************                                                   NEW10500
C          OUTPUT OPTIONS                                                     NEW10510
C          ****************                                                   NEW10520
           IF((PASS.EQ.3).OR.(PASS.EQ.4))THEN                                 NEW10530
            END=2*ORDER                                                       NEW10540
           ENDIF                                                              NEW10550
           IF((PASS.EQ.1).OR.(PASS.EQ.2))THEN                                 NEW10560
            END=ORDER                                                         NEW10570
           ENDIF                                                              NEW10580
C          MAKE DENOMINATOR COEFFICIENT OF Z**END EQUAL TO 1                  NEW10590
           DO 250 M=1,2                                                       NEW10600
            DO 251 N=0,END                                                    NEW10610
             DFLTR(M,N)=DFLTR(M,N)/DFLTR(2,END)                               NEW10620
251         CONTINUE                                                          NEW10630
250        CONTINUE                                                           NEW10640
           CALL EXCMS ('CLRSCRN')                                             NEW10650
           WRITE(6,*)                                                         NEW10660
           WRITE(6,*) 'THE FINAL DIGITAL FILTER IS:'                          NEW10670
           WRITE(6,*)                                                         NEW10680
           WRITE(6,*) '    NUMERATOR COEFFICIENTS:'                           NEW10690
           DO 300 I=END,0,-1                                                  NEW10700
            WRITE(6,902) DFLTR(1,I),I                                         NEW10710
300        CONTINUE                                                           NEW10720
           WRITE(6,*) '    DENOMINATOR COEFFICIENTS:'                         NEW10730
           DO 301 J=END,0,-1                                                  NEW10740
            WRITE(6,902) DFLTR(2,J),J                                         NEW10750
301        CONTINUE                                                           NEW10760
           WRITE(6,*)                                                         NEW10770
           WRITE(6,*) ' ** ENTER 1 TO CONTINUE **'                            NEW10780
           READ(5,*) LOOK                                                     NEW10790
C          ***************************************************************** NEW10800
```

FILE: DFCADD FORTRAN A

```
C     OPTION TO SPOT CHECK FILTER OUTPUT MAGNITUDE AND FREQUENCY          NEW10810
      CALL EXCMS ('CLRSCRN')                                              NEW10820
      WRITE(6,*) 'THE TRANSFER FUNCTION FOR THIS FILTER HAS BEEN'         NEW10830
      WRITE(6,*) 'COMPUTED. DO YOU WANT TO SPOT CHECK THE MAGNITUDE'      NEW10840
      WRITE(6,*) 'AND PHASE OF THE FILTER OUTPUT FOR A GIVEN '            NEW10850
      WRITE(6,*) 'FREQUENCY?'                                             NEW10860
      WRITE(6,*) '    ** 1-YES/2-NO ** '                                  NEW10870
      READ(5,*)  LOOK                                                     NEW10880
      IF(LOOK.EQ.1)THEN                                                   NEW10890
310    CALL EXCMS ('CLRSCRN')                                             NEW10900
       IF(AFORDF.EQ.1)THEN                                               NEW10910
         WRITE(6,*) 'WHAT ANALOG FREQUENCY(IN KHZ)'                      NEW10920
         WRITE(6,*) ' DO YOU WANT TO CHECK?'                             NEW10930
         WRITE(6,*) '   ** DECIMAL **'                                   NEW10940
         READ(5,*)  F                                                    NEW10950
         DF=(F/SSFREQ)*2.0*SPI                                           NEW10960
       ENDIF                                                             NEW10970
       IF(AFORDF.EQ.2)THEN                                               NEW10980
         WRITE(6,*) 'WHAT DIGITAL FREQUENCY DO YOU WANT TO CHECK?'       NEW10990
         WRITE(6,*) '  ** DECIMAL ** '                                   NEW11000
         READ(5,*) DF                                                    NEW11010
       ENDIF                                                             NEW11020
       X=COS(DF)                                                         NEW11030
       Y=SIN(DF)                                                         NEW11040
       Z=CMPLX(X,Y)                                                      NEW11050
       XN=REAL(DFLTR(1,0))                                               NEW11060
       YN=0.0                                                            NEW11070
       XD=REAL(DFLTR(2,0))                                               NEW11080
       YD=0.0                                                            NEW11090
       HOFZN=CMPLX(XN,YN)                                                NEW11100
       HOFZD=CMPLX(XD,YD)                                                NEW11110
       DO 320 N=1,END                                                    NEW11120
        ZN=REAL(DFLTR(1,N))*(Z**N)                                       NEW11130
        ZD=REAL(DFLTR(2,N))*(Z**N)                                       NEW11140
        HOFZN=HOFZN+ZN                                                   NEW11150
        HOFZD=HOFZD+ZD                                                   NEW11160
320    CONTINUE                                                          NEW11170
       HOFZ=HOFZN/HOFZD                                                  NEW11180
       FMAG=SQRT((REAL(HOFZ)**2)+(AIMAG(HOFZ)**2))                       NEW11190
       FMAGDB=20.0*ALOG10(FMAG)                                          NEW11200
       FPHSE=ATAN(AIMAG(HOFZ)/REAL(HOFZ))                                NEW11210
       WRITE(6,*) 'THE FILTER OUTPUT WILL BE:'                           NEW11220
       WRITE(6,914) FMAGDB                                               NEW11230
       WRITE(6,915) FPHSE                                                NEW11240
       WRITE(6,*)                                                        NEW11250
       WRITE(6,*) 'DO YOU WANT TO CHECK ANOTHER FREQUENCY?'              NEW11260
       WRITE(6,*) '    ** 1-YES/2-NO ** '                                NEW11270
       READ(5,*) LOOK                                                    NEW11280
       IF(LOOK.EQ.1)GO TO 310                                            NEW11290
      ENDIF                                                              NEW11300
C     ****************************************************************  NEW11310
C     OPTION TO LOAD FREQUENCY RESPONSE INTO A DATA FILE                 NEW11320
      CALL EXCMS ('CLRSCRN')                                             NEW11330
      WRITE(6,*) 'DO YOU WANT THE FILTERS FREQUENCY RESPONSE '           NEW11340
      WRITE(6,*) 'DATA TO BE ENTERED INTO A DATA FILE FOR USE'           NEW11350
      WRITE(6,*) 'BY A PLOTTING PROGRAM? '                               NEW11360
      WRITE(6,*) ' NOTE: THE FILE NAME WILL BE FILTER, FILE TYPE'        NEW11370
      WRITE(6,*) '     WILL BE DATA, 100 POINTS WILL BE CALCULATED'      NEW11380
      WRITE(6,*) '     FOR A DIGITAL FREQUENCY FROM 0 TO PI, THE FIRST' NEW11390
      WRITE(6,*) '     COLUMN WILL BE THE DIGITAL FREQUENCY (EXPRESSED 'NEW11400
      WRITE(6,*) '     AS A FRACTION OF THE SAMPLING FREQUENCY 0-1/2) ' NEW11410
      WRITE(6,*) '     THE SECOND COLUMN WILL BE THE MAGNITUDE, AND'     NEW11420
      WRITE(6,*) '     THE THIRD COLUMN WILL BE THE PHASE(DEGREES)'      NEW11430
      WRITE(6,*) '     ** 1-MAKE DATA FILE/2-DON''T BOTHER **'           NEW11440
      READ(5,*)  LOOK                                                    NEW11450
      IF(LOOK.EQ.1)THEN                                                  NEW11460
       DO 350 I=0,100                                                    NEW11470
        DF=REAL(I)*(5.0/1000.0)                                          NEW11480
        DFR=REAL(I)*(SPI/100.0)                                          NEW11490
        X=COS(DFR)                                                       NEW11500
        Y=SIN(DFR)                                                       NEW11510
        Z=CMPLX(X,Y)                                                     NEW11520
```

FILE: DFCADD   FORTRAN A

```
        XN=REAL(DFLTR(1,0))                                                    NEW11530
        YN=0.0                                                                 NEW11540
        XD=REAL(DFLTR(2,0))                                                    NEW11550
        YD=0.0                                                                 NEW11560
        HOFZN=CMPLX(XN,YN)                                                     NEW11570
        HOFZD=CMPLX(XD,YD)                                                     NEW11580
        DO 351 N=1,END                                                         NEW11590
         ZN=REAL(DFLTR(1,N))*(Z**N)                                            NEW11600
         ZD=REAL(DFLTR(2,N))*(Z**N)                                            NEW11610
         HOFZN=HOFZN+ZN                                                        NEW11620
         HOFZD=HOFZD+ZD                                                        NEW11630
351     CONTINUE                                                               NEW11640
        HOFZ=HOFZN/HOFZD                                                       NEW11650
        FMAG=SQRT((REAL(HOFZ)**2)+(AIMAG(HOFZ)**2))                            NEW11660
        IF(REAL(HOFZ).NE.0.0)THEN                                              NEW11670
         FPHSER=ATAN(AIMAG(HOFZ)/REAL(HOFZ))                                   NEW11680
        ELSEIF(REAL(HOFZ).EQ.0.0)THEN                                          NEW11690
         IF(AIMAG(HOFZ).GT.0.0)FPHSER=1.570796                                 NEW11700
         IF(AIMAG(HOFZ).LT.0.0)FPHSER=-1.570796                                NEW11710
         IF(AIMAG(HOFZ).EQ.0.0)FPHSER=0.0                                      NEW11720
        ENDIF                                                                  NEW11730
        FPHSE=FPHSER*(180.0/SPI)                                               NEW11740
        WRITE(8,916) DF,FMAG,FPHSE                                             NEW11750
350    CONTINUE                                                                NEW11760
      ENDIF                                                                    NEW11770
C     ***************************************************************          NEW11780
901   FORMAT(5X,F10.5,' S**',I1)                                               NEW11790
902   FORMAT(5X,F10.5,' Z**',I1)                                               NEW11800
903   FORMAT(3X,'ALPHA= ',F9.4)                                                NEW11810
904   FORMAT(3X,'K= ',F9.4)                                                    NEW11820
905   FORMAT(3X,'THE LOWPASS DIGITAL PROTOTYPE FREQ. IS  ',F5.3)               NEW11830
914   FORMAT(3X,'MAGNITUDE =  ',F8.2,' DB')                                    NEW11840
915   FORMAT(3X,'PHASE =  ',F6.2,' RADIANS')                                   NEW11850
916   FORMAT(3X,F4.3,10X,F5.3,10X,F6.2)                                        NEW11860
      STOP                                                                     NEW11870
      END                                                                      NEW11880
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	1
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
3.	Chairman, Code 62 Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4.	Commandant of the Marine Corps Code TE 06 Headquarters, U.S. Marine Corps Washington, DC 20380-0001	1
5.	United States Military Academy Department of Electrical Engineering ATTN: MAJ E. Siomacco West Point, New York 10996	1
6.	Department of Electrical and Computer Engineering ATTN: Prof. R. Strum, Code 62St Naval Postgraduate School Monterey, California 93943-5000	1
7.	Department of Electrical and Computer Engineering ATTN: Prof. J. England, Code 62 Naval Postgraduate School Monterey, California 93943-5000	1
8.	Department of Electrical and Computer Engineering ATTN: Prof. P. Moose, Code 62Me Naval Postgraduate School Monterey, California 93943-5000	1
9.	Department of Electrical and Computer Engineering ATTN: Prof. D. Kirk, Code 62 Naval Postgraduate School Monterey, California 93943-5000	1

321-897

Thesis
R37825 Rich
c.1 Algorithms for computer aided design of digital filters.

Thesis
R37825 Rich
c.1 Algorithms for computer aided design of digital filters.

AMERICAN BINDERY
TOPEKA KANSAS

thesR37285
c.1
Algorithms for computer aided design of
3 2768 000 81718 3
DUDLEY KNOX LIBRARY